

# Four-Round Concurrent Non-Malleable Commitments from One-Way Functions<sup>\*</sup>

Michele Ciampi<sup>1</sup>, Rafail Ostrovsky<sup>2</sup>, Luisa Siniscalchi<sup>1</sup>, and Ivan Visconti<sup>1</sup>

<sup>1</sup> DIEM, University of Salerno, ITALY  
{mciampi,lsiniscalchi,visconti}@unisa.it

<sup>2</sup> UCLA, USA  
rafail@cs.ucla.edu

**Abstract.** How many rounds and which assumptions are required for *concurrent* non-malleable commitments? The above question has puzzled researchers for several years. Pass in [TCC 2013] showed a lower bound of 3 rounds for the case of black-box reductions to falsifiable hardness assumptions with respect to polynomial-time adversaries. On the other side, Goyal [STOC 2011], Lin and Pass [STOC 2011] and Goyal et al. [FOCS 2012] showed that one-way functions (OWFs) are sufficient with a constant number of rounds. More recently Ciampi et al. [CRYPTO 2016] showed a 3-round construction based on subexponentially strong one-way permutations. In this work we show as *main result* the first 4-round concurrent non-malleable commitment scheme assuming the existence of any one-way function.

Our approach builds on a new security notion for argument systems against man-in-the-middle attacks: *Simulation-Witness-Independence*. We show how to construct a 4-round one-many simulation-witnesses-independent argument system from one-way functions. We then combine this new tool in parallel with a weak form of non-malleable commitments constructed by Goyal et al. in [FOCS 2014] obtaining the main result of our work.

## 1 Introduction

Commitment schemes are a fundamental primitive in Cryptography. Here we consider the intriguing question of constructing round-efficient schemes that remain secure even against man-in-the-middle (MiM) attacks: non-malleable (NM) commitments [12].

**Non-malleable commitments.** The round complexity of commitment schemes in the stand-alone setting is nowadays well understood. Non-interactive commitments can be constructed assuming the existence of 1-to-1 one-way functions

---

<sup>\*</sup> The final authenticated version is available online at [https://doi.org/10.1007/978-3-319-63715-0\\_5](https://doi.org/10.1007/978-3-319-63715-0_5)

(OWFs) [18]; 2-round commitments can be constructed assuming the existence of OWFs only. Moreover non-interactive commitments do not exist if one relies on the black-box use of OWFs only [33].

Instead, the round complexity of NM commitments<sup>3</sup> after 25 years of research remains a fascinating open question, in particular when taking into account the required computational assumptions. The original construction of [12] required a logarithmic number of rounds and the sole use of OWFs. Then, through a long sequence of very exciting positive results [1, 41, 43, 42, 46, 45, 31, 47, 51, 29, 30, 19, 22], the above open question has been in part solved obtaining a constant-round<sup>4</sup> (even concurrent) NM commitment scheme by using any OWF in a black-box fashion. On the negative side, Pass proved that NM commitments require at least 3 rounds [40]<sup>5</sup> when security is proved through a black-box reduction to polynomial-time hardness assumptions.

**Breaking the multiple rewind-slot barrier.** The above papers left open the question of achieving (concurrent) non-malleable commitments with optimal round complexity. A main common issue for round-efficient non-malleable commitments is that typically a security proof requires some simulation on the left and extraction on the right that should not interfere with each other. Indeed, a known paradigm introduced by Pass [39] proposes to have in a protocol multiple potential rewind slots so that extraction and simulation can both be run in 2 independent sequential steps. On the negative side, the use of multiple rewind slots increases the round complexity of the protocol (i.e., two rewind slots require at least 5 rounds).

More recently the multiple rewind-slot technique has been bypassed in [25] but only for the (simpler) one-one case (i.e., just one sender and one receiver). In particular, Goyal et al. [25] showed a *one-one* 4-round NM commitment scheme based on OWFs only. The more recent work of Goyal et al. [24, 23] exploited the use of the NM codes in the split-state model to show a 3-round one-one NM commitment scheme based on the black-box use of any 1-to-1 OWF that is secure against super-polynomial time adversaries<sup>6</sup>. Ciampi et al. [6] obtained concurrent non-malleability in 3 rounds starting from any one-one non-malleable (and extractable) commitment scheme, but their security proof crucially relies on the existence of one-way permutations secure against subexponential-time adversaries. Assumptions against super-polynomial time adversaries allow to avoid

---

<sup>3</sup> In this paper we will consider only NM commitments w.r.t. commitments. For the case of NM w.r.t. decommitments see [43, 46, 37, 2, 9, 21].

<sup>4</sup> The construction of [22] can be compressed to 6 rounds (see [25]).

<sup>5</sup> If instead one relies on non-standard assumptions or trusted setups (e.g., using trusted parameters, working in the random oracle model, relying on the existence of NM OWFs) then there exist non-interactive NM commitments [10, 38].

<sup>6</sup> While [24, 23] only claimed one-one non-malleability, the difficulty of achieving concurrent non-malleability was discussed in [6] where Ciampi et al. showed an explicit successful concurrent man-in-the-middle for the preliminary eprint version of [24].

multiple rewind slots even in presence of polynomially many sessions since the security proof can rely on straight-line simulation/extraction<sup>7</sup>.

### 1.1 Our Results

In this paper we break the multiple-slot barrier for concurrent NM commitments by showing a 4-round scheme based on the sole existence of OWFs. While previous work relied on having either 1) stronger assumptions or 2) multiple rewind slots or 3) limited concurrency, in this work we introduce new techniques that allow to have just one rewind slot, minimal hardness assumptions and full concurrency.

More specifically we give the following four contributions.

**Non-malleable commitments w.r.t. non-aborting adversaries.** We prove that a subprotocol of [25] is a 4-round statistically binding concurrent NM commitment scheme from OWFs (resp. a 3-round perfectly binding concurrent NM commitment scheme from 1-to-1 OWFs), if the adversary is restricted to playing well-formed commitments in the right sessions when receiving well formed commitments from the left sessions. We refer to this weaker security notion as concurrent weak non-malleability (wNM).

**Simulation-Witness-Independence.** We define a new security notion for argument systems w.r.t. man-in-the-middle attacks that we refer to as simulation-witness-independence (SimWI). This security notion seemingly is not implied by previous notions as simulation-extractability/soundness and strong non-malleable witness indistinguishability.

**4-Round One-Many SimWI from OWFs.** We then construct a 4-round one-many SimWI argument of knowledge for some specific languages by relying on OWFs only. This construction circumvents the major problem caused by the need of rewinding on the left to simulate and on the right to extract when there is only one available rewind slot.

**Concurrent wNM + One-Many SimWI  $\Rightarrow$  4-Round Concurrent NM Commitments.** We present our new paradigm consisting in combining the above two notions in a protocol that runs in parallel the concurrent wNM commitment scheme and the one-many SimWI argument of knowledge. Therefore as main result of this work we upgrade concurrent wNM to full-fledged concurrent non-malleability without any penalization in rounds and assumptions.

We now discuss in more details each of the above 4 contributions.

**Weak Non-Malleable Commitments** We define commitment schemes enjoying a limited form of non-malleability<sup>8</sup>.

<sup>7</sup> Hardness assumptions against subexponential-time adversaries were already used in [41, 47, 51] to improve the round-complexity of NM commitments.

<sup>8</sup> We remark that Goyal in [19] defined a weaker notion of non-malleable commitments (non-malleability w.r.t. replacement) that also had the goal to deal with commit-

Informally, we say that a commitment scheme is weak non-malleable (wNM) if it is non-malleable w.r.t. adversaries that never commit to  $\perp$  when receiving honestly computed commitments. This form of non-malleability is significantly weaker than full-fledged non-malleability. Indeed, a full-fledged MiM  $\mathcal{A}$  can for instance maul as follows:  $\mathcal{A}$  creates a commitment of  $m_0$  making use of messages computed by the sender in the left session so that if the sender commits to  $m_0$  then the commitment of  $\mathcal{A}$  is a well formed commitment of  $m_0$ , while instead if the sender commits to  $m_1 \neq m_0$  then the commitment of  $\mathcal{A}$  is not well formed and therefore corresponds to  $\perp$ . Such attacks can be explicitly instantiated as shown in [6] where a generalization of the above  $\mathcal{A}$  is used to prove that a preliminary version of the scheme of [24] is not concurrent non-malleable.

While by itself the wNM guarantee is certainly unsatisfying as protection against MiM attacks, the design of a wNM commitment scheme can be an easier task and schemes with such light non-malleability flavor might exist with improved round complexity, efficiency and complexity assumptions compared to schemes achieving full-fledged non-malleability.

We show that a protocol due to [25] is a 4-round statistically binding concurrent wNM commitment scheme requiring OWFs only (resp., a 3-round perfectly binding concurrent wNM commitment scheme requiring 1-to-1 OWFs only). Moreover their protocol can be instantiated to be public coin. The security proof consists of some pretty straightforward observations on top of various useful lemmas already proven in [25]. Our contribution on wNM commitments therefore consists in 1) introducing and formalizing this notion; 2) observing the existence of a secure construction in previous work; 3) using it as one of the two main building blocks of our paradigm allowing to obtain 4-round concurrent (full-fledged) NM commitments from OWFs. For lack of space we postpone further details on wNM commitments to the full version (see [5]) so that in this work we can give more details on the more interesting results of this work (i.e., the definition and construction of SimWI, and the new paradigm for concurrent NM commitments). A formal definition of weak NM commitments can be found in Sec. 2.3 (see Def. 5). The proof that a scheme proposed in [25] satisfies this notion can be found in the full version (see [5]).

**Simulation-Witness-Independence.** We introduce a new security notion against MiM attacks to argument systems. We call our security notion *simulation-witness-independence* (SimWI) since it has similarities both with simulation extractability/soundness (see [44, 49]) and with (strong) non-malleable witness indistinguishability [32, 36] (sNMWI, NMWI). For simplicity we will discuss now the case of one prover and one verifier only, however our formal definition, construction and application will focus on the one-many case (i.e., up to 1 prover and polynomially many verifiers).

The 1st security flavor that our notion tries to capture is the concept that the view of a MiM in the real game should be simulatable. Therefore we will

---

ments of  $\perp$ . While the goal is similar to our definition, the actual formulation is quite different.

have an experiment corresponding to the real game where the MiM plays with a honest prover and a honest verifier, and an experiment corresponding to the simulated game that simply consists of the output of a stand-alone simulator that emulates the prover and runs the code of honest verifiers when interacting internally with the MiM<sup>9</sup>.

While the above 1st security flavor guarantees that the statements proven by the MiM in the real-world experiment and in the simulated experiment are indistinguishable, there still is no guarantee that the MiM is unable to prove in the two experiments statements that are associated to witnesses belonging to distinguishable distributions. In other words, as 2nd flavor we want to capture the independence of the witnesses associated to the statements proven by the MiM with respect to the fact that the actual witness in the left session is used (this is the case of the real game) or is not used (this is the case of the simulated game). In order to avoid any ambiguity on which witness is associated to a statement, we associate to an  $\mathcal{NP}$  language a non-negative integer  $\gamma$ . More precisely, for any  $\mathcal{NP}$  language  $L$  we consider a non-negative integer  $\gamma$  such that for any  $x \in L$  all witnesses of  $x$  have the same first  $\gamma$  bits<sup>10</sup>. The reason why we assign such a value  $\gamma$  to every  $\mathcal{NP}$  language is that it fixes in some non-ambiguous way the input for the distinguisher of SimWI (indeed the input will be the first  $\gamma$  bits of any witness) and at same time the prefix of all the witnesses of an instance can be recovered by extracting any witness.

The above 2nd flavor makes our security definition non-trivial. Indeed standard zero knowledge is clearly insufficient against MiM attacks and the definition has strong connections with the (hard to achieve) concept of committed message in NM commitments<sup>11</sup>. One might think that some heavy machinery could already imply our new notion. However, by taking into account all subtleties of the definitions it turns out that SimWI is seemingly not implied by simulation extractability, simulation soundness and sNMWI/NMWI. We stress that our goal is to get a *one-many* 4-round construction under minimal assumptions.

**Comparison with simulation extractability and simulation soundness.** Simulation extractability requires the simulator to output a transcript and witnesses for the statements appearing in the right sessions of the transcript.

Simulation soundness requires the MiM to fail in proving false statements when receiving simulated proofs of false statements.

SimWI requires the simulator to output a transcript that includes statements proven in right sessions. The distribution of the instance/witness pairs associ-

<sup>9</sup> There is nothing surprising so far, this is just the concept of zero knowledge naturally augmented by extending the simulator with the behavior of honest verifiers to feed the MiM with messages belonging to the right sessions too.

<sup>10</sup> Note that when  $\gamma = 0$  the 2nd security flavor is cancelled and SimWI becomes equivalent to zero knowledge. Furthermore when  $\gamma$  is equal to the largest witness size then we are considering languages in  $\mathcal{UP}$ .

<sup>11</sup> We stress that the main goal of this work is to construct 4-round concurrent NM commitments from OWFs, and we will achieve it by making use of SimWI. As such, to avoid circularity, we can not use concurrent NM commitments to construct SimWI.

ated to those statements is required to be indistinguishable from the distribution of the instance/witness pairs associated to the statements proved by the MiM in the real game. In simulation extractability there is no requirement on the witness given in output by the simulator beyond being valid witnesses. Simulation soundness does not have any requirement on the witnesses associated to the statements proven by the MiM.

**Comparison with sNMWI/NMWI.** sNMWI considers two indistinguishable distributions of instance/witnesses pairs. Very informally, the requirement of sNMWI/NMWI is that the instance/witness pairs associated to the arguments given by the MiM in the right sessions be independent of the distribution from which the instance/witness pair of the argument given to the MiM in the left session has been sampled.

SimWI requires the existence of a simulator while instead sNMWI/NMWI only considers experiments where the actual prover plays.

**One-Many SimWI From OWFs in 4 Rounds (i.e., in Just One Rewind Slot!).** As discussed above, SimWI is an interesting security notions w.r.t. MiM attacks and similarly to all previous non-malleability notions is certainly non-trivial to achieve, especially when considering 1) the one-many case 2) only four rounds (i.e., one rewind slot) and 3) minimal assumptions. In this work we show how to construct a 4-round one-many SimWI argument of knowledge (AoK) from OWFs, therefore avoiding multiple rewind slots.

A common approach to construct 4-round zero-knowledge arguments (even without non-malleability requirements) relies on the FLS/FS paradigm [14, 15]. First there is a subprotocol useful to extract a trapdoor from the adversarial verifier. Then there is a witness-indistinguishable proof of knowledge (WIPoK) where the prover proves knowledge of either a witness for the statement or of the trapdoor. In order to save rounds the two subprotocols are parallelized.

The above common approach fails in presence of MiM attacks. The reason is that the MiM adversary can attack the witness indistinguishability (WI) of the WIPoK received in the left session in order to prove his statements in the right sessions. Using such a MiM to contradict the WI of the WIPoK is problematic since one should extract some useful information from the right session but this would require also to rewind the challenger of the WI of the WIPoK on the left.

We bypass the above difficulty as follows. Instead of relying on the WI of the WIPoK that requires two messages played by the challenger, we propose a construction where we essentially break the interactive challenger of WI into two non-interactive challengers. We implement this idea by relying on: 1) instance-dependent trapdoor commitments (IDTCom) and 2) special honest-verifier zero knowledge (special HVZK). More in details, let  $(\pi_1, \pi_2, \pi_3, \pi_4)$  be the transcript of a delayed-input<sup>12</sup> 4-round special HVZK adaptive-input proof of knowledge (PoK). We require the prover to send an IDTCom com of  $\pi_2$  that is opened,

<sup>12</sup> By *delayed-input* we mean that the statement will be known only at the last round. The delayed-input property has been critically used in the past (e.g., [27, 11, 52]) and very recently (e.g., [7, 8, 16, 6, 26, 34]), since it helps in improving the round complexity of external protocols.

sending the opening  $\text{dec}$ , only in the last round, when  $\pi_4$  is sent. The actual transcript therefore becomes  $(\pi_1, \text{com}, \pi_3, (\pi_2, \text{dec}, \pi_4))$ .

Consider now an experiment where the trapdoor is known and  $\pi_2$  can be opened arbitrarily. If the output of the experiment deviates from the original one, we will have a reduction to the trapdooriness of the IDTCom. The reduction is not problematic since the challenger of the trapdooriness is non-interactive, sending a pair (commitment, decommitment) that is either computed using the regular procedure or through the use of the trapdoor. Next, in another experiment we can replace the prover of the PoK with the special HVZK simulator that will compute  $\pi_2$  and  $\pi_4$  after having as input  $\pi_1$  and  $\pi_3$ . Again, the output of this experiment will not deviate from the previous one otherwise we can show an adversary for the special HVZK property. The reduction again is not problematic since the challenger of special HVZK is non-interactive.

We implement the trapdoor-extraction subprotocol through OWFs by using as trapdoor knowledge of two signatures under the same public key sent by the verifier in the 1st round. The verifier will send a signature of one message (chosen by the prover) in the 3rd round (with this approach we follow previous ideas of [11, 20, 4, 3]). We will use a delayed-input special HVZK adaptive-input PoK where the prover proves knowledge of either a witness for the statement or of signatures of messages. The IDTCom will have the public key of the signature scheme as instance, therefore the simulator after having extracted the signatures will be able to equivocate the commitments. The security proof presents one more caveat. Once the simulator rewinds on the left to obtain the trapdoor it is not clear how to argue that the extraction from the right is meaningful since the extractor might simply obtain the same trapdoor. More specifically, the adversary might be able to equivocate on the right, therefore the extractor of the PoK would fail, and the best we can get from such a binding violation is the trapdoor of the IDTCom played in the right session. This does not give any contradiction since the trapdoor of the right session had to be already known in order to answer twice (before and after the rewind) in the right session to the MiM. We resolve this problem by relying on a specific proof approach where while the initial transcript is generated by the simulator, when the extractions are played in the right sessions, the transcript of the left session is re-completed by running the prover of the special HVZK PoK. The reason why in this case the extraction on the right will succeed is that if we extract the trapdoor from the right session then this will also happen in the real game where the trapdoor is never used. In turn it would break the security of the signature scheme.

**Caveat: adaptive-input selection.** We will give a formal definition that allows the MiM to select the instance/witness pair for the left session only at the end, while the MiM must fix the statement for a right session already when playing his first round in that session. Our construction satisfies this notion and even a more important form of adaptiveness. We allow the MiM to specify the statement in the last round of a right session, as long as the witness is already fixed when playing his first round in that session. The reason why we prove such more sophisticated form of adaptive-input selection is that it is required



in our application for concurrent NM commitments. Ideally one would like to satisfy the best possible adaptive-input selection, in order to make this new primitive useful in a broader range of applications. However we can not prove our construction secure with fully adaptive-input selection since we are not able to extract the witness from a MiM selecting a new statement (with possibly a new witness) in the last round of a right session. Indeed we would end up having a certain statement in the transcript of the simulator and then a witness for another statement obtained through rewinds. This would negatively affect our proof approach.

**4-Round Concurrent NM Commitments from OWFs.** We solve the problem left open by [25] by showing a 4-round concurrent NM commitment scheme relying on OWFs only. The new paradigm that we propose to obtain concurrent non-malleability consists in combining in parallel a 4-round public-coin concurrent wNM commitment scheme from OWFs  $\Pi_0$ , and a one-many 4-round SimWI argument of knowledge from OWFs  $\Pi_1$ .

**The new paradigm.**  $\Pi_0$  is run in order to commit to the message  $m$ .  $\Pi_1$  is instead used to prove knowledge of a valid message and randomness explaining the transcript of  $\Pi_0$ . The power of the new approach consists in using the above two tools that are in perfect synergy to defeat a concurrent MiM attack. The idea of the security proof is now quite simple. Since any one-many NM commitment is also many-many<sup>13</sup> NM, we focus the following discussion on the one-many case.

In the 1st experiment (the real game RG0) the sender commits to  $m_0$ . Clearly there can not be a commitment to  $\perp$  on the right otherwise the soundness of  $\Pi_1$  is contradicted. Symmetrically there is an experiment RG1 where the sender commits to  $m_1$  and there is no commitment to  $\perp$  on the right. Then we consider an hybrid game H0 where the simulator of one-many SimWI of  $\Pi_1$  is used. Observe that if (by contradiction) the distribution of the messages committed on the right changes w.r.t. RG0 we have that also the distribution of the witnesses corresponding to the statements proved in  $\Pi_1$  on the right changes. However this clearly violates SimWI. Therefore it must still be the case that a commitment played on the right corresponds to  $\perp$  with negligible probability only. Symmetrically, there is an experiment H1 that is indistinguishable from RG1 and such that commitments played on the right are well formed (i.e., different from  $\perp$ ). Therefore we can conclude that RG0 is indistinguishable from RG1 by noticing that H0 is indistinguishable from H1. Indeed, both H0 and H1 guarantee that the messages committed by the adversary on the right correspond to  $\perp$  with negligible probability only. Summing up, a detectable deviation from H0 to H1 implies a contradiction of the concurrent wNM of  $\Pi_0$ <sup>14</sup>. This observation concludes the high-level overview of the security proof. However, some remarks are in order.

<sup>13</sup> A many-many NM commitment scheme can be also indicate as a concurrent NM commitment scheme. In the rest of the paper we use the term concurrent.

<sup>14</sup> This reduction needs extra help, see Remark 2 below.



**Remark 1: the required adaptive-input flavor.** As specified in the previous section, our 4-round one-many SimWI AoK  $\Pi_1$  is fully adaptive on the left but instead on the right requires the witness to be fixed already in the first round of the MiM. The statements instead can be decided in the last round also in the right sessions. The flexibility with the statement is important since  $\Pi_0$  is completed only in the last round and the entire transcript of  $\Pi_0$  is part of the statement of  $\Pi_1$ . The lack of flexibility on the witness in the right sessions forces us to add one more requirement to  $\Pi_0$ . We need that message and randomness are already fixed in the 2nd round of  $\Pi_0$ , since they will be the witness for  $\Pi_1$ . This property is satisfied by the construction of [25] that we prove to be concurrent wNM in the full version (see [5]).

**Remark 2: on the need of public coins in  $\Pi_0$ .** In a reduction we will have to simulate the last round of the receiver of  $\Pi_0$  without knowing the randomness he used to compute the previous round. Obviously public coins are easy to simulate.

## 1.2 3-Round Concurrent Non-Malleable Commitments

The work of Ciampi et al. [6] relied on subexponentially strong one-way permutations and the existence of any 3-round one-one non-malleable commitment scheme  $\Pi$ . In this work we propose a different approach for 3-round one-one non-malleable commitments that instead can start with a limited form of non-malleability enjoyed by both a subprotocol of [25] and a subprotocol of [24] (therefore we can instantiate our result in two completely different ways). The result of Ciampi et al. [6] can still be instantiated using our 3-round one-one non-malleable commitment scheme that we present in this work. Therefore our work combined with the one of [6] gives the first 3-round concurrent non-malleable commitment scheme from falsifiable assumptions<sup>15</sup>.

Our 3-round one-one non-malleable commitment scheme combines some ideas of [6] along with the concept of weak non-malleable commitment. In particular we start with a scheme that is one-one non-malleable only against synchronous adversaries that do not commit to  $\perp$ . As we discuss in the paper, both a subprotocol of [25] and a subprotocol of [24] satisfy this security property. Considering this notion we construct a compiler that, on input a 3-round synchronous weak one-one NM commitment scheme, gives as output a 3-round extractable one-one NM commitment scheme assuming OWPs secure against subexponential-time adversaries. This can then be used inside [6] to get 3-round concurrent non-malleable commitments from subexponential one-way permutations.

<sup>15</sup> We stress that after our results were publicly available, a construction for 3-round one-one non-malleable commitments with the black-box use of one-to-one one-way functions secure against quasi-polynomial-time adversaries was announced in [23]. Their work revisited the primitives that instantiated their prior construction [24] based on 1-1 OWFs that appeared before this work and before [6].

### 1.3 The New State of the Art

In Table 1 we summarize the new state of the art.

Paper	No. Rounds	Assumption	Concurrency	
Goyal/ Lin and Pass	STOC 2011	$\geq 6$	OWFs	Yes
Goyal et al.	FOCS 2012	$\geq 6$	BB OWFs	Yes
Goyal et al.	FOCS 2014	4	OWFs	No
This work + Ciampi et al.	CRYPTO 2016	3	subexp OWPs	Yes
This work (main result)		4	OWFs	Yes
Goyal et al.	STOC 2016 <sup>16</sup>	3	BB quasi-poly 1-1 OWFs	No

Table 1: Comparison with recent positive results from the oldest to the newest.

## 2 Definitions and Tools

### 2.1 Preliminaries

We denote the security parameter by  $\lambda$  and use “|” as concatenation operator (i.e., if  $a$  and  $b$  are two strings then by  $a|b$  we denote the concatenation of  $a$  and  $b$ ). For a finite set  $Q$ ,  $x \leftarrow Q$  sampling of  $x$  from  $Q$  with uniform distribution. We use the abbreviation PPT that stands for probabilistic polynomial time. We use  $\text{poly}(\cdot)$  to indicate a generic polynomial function and  $\mathbb{N}$  to denote the set of positive integer. We use the notation  $s^t$  to denote the first  $t$ -bits of a string  $s$ . A *polynomial-time relation*  $\text{Rel}$  (or *polynomial relation*, in short) is a subset of  $\{0, 1\}^* \times \{0, 1\}^*$  such that membership of  $(x, w)$  in  $\text{Rel}$  can be decided in time polynomial in  $|x|$ . For  $(x, w) \in \text{Rel}$ , we call  $x$  the *instance* and  $w$  a *witness* for  $x$ . For a polynomial-time relation  $\text{Rel}$ , we define the  $\mathcal{NP}$ -language  $L_{\text{Rel}}$  as  $L_{\text{Rel}} = \{x | \exists w : (x, w) \in \text{Rel}\}$ . Analogously, unless otherwise specified, for an  $\mathcal{NP}$ -language  $L$  we denote by  $\text{Rel}_L$  the corresponding polynomial-time relation (that is,  $\text{Rel}_L$  is such that  $L = L_{\text{Rel}_L}$ ). Let  $A$  and  $B$  be two interactive probabilistic algorithms. We denote by  $\langle A(\alpha), B(\beta) \rangle(\gamma)$  the distribution of  $B$ 's output after running on private input  $\beta$  with  $A$  using private input  $\alpha$ , both running on common input  $\gamma$ . Typically, one of the two algorithms receives  $1^\lambda$  as input. A *transcript* of  $\langle A(\alpha), B(\beta) \rangle(\gamma)$  consists of the messages exchanged during an execution where  $A$  receives a private input  $\alpha$ ,  $B$  receives a private input  $\beta$  and both  $A$  and  $B$  receive a common input  $\gamma$ . Moreover, we will refer to the *view* of  $A$  (resp.  $B$ ) as the messages it received during the execution of  $\langle A(\alpha), B(\beta) \rangle(\gamma)$ , along with its randomness and its input.

<sup>16</sup> The need of super-polynomial time hardness assumptions appeared in December 2016 [23].

**Definition 1 (Proof/argument system).** A pair of PPT interactive algorithms  $\Pi = (\mathcal{P}, \mathcal{V})$  constitutes a proof system (resp., an argument system) for an  $\mathcal{NP}$ -language  $L$ , if the following conditions hold:

**Completeness:** For every  $x \in L$  and  $w$  such that  $(x, w) \in \text{Rel}_L$ , it holds that:  $\text{Prob}[\langle \mathcal{P}(w), \mathcal{V} \rangle(x) = 1] = 1$ .

**Soundness:** For every interactive (resp., PPT interactive) algorithm  $\mathcal{P}^*$ , there exists a negligible function  $\nu$  such that for every  $x \notin L$  and every  $z$ :  $\text{Prob}[\langle \mathcal{P}^*(z), \mathcal{V} \rangle(x) = 1] < \nu(|x|)$ .

A proof/argument system  $\Pi = (\mathcal{P}, \mathcal{V})$  for an  $\mathcal{NP}$ -language  $L$ , enjoys *delayed-input* completeness if  $\mathcal{P}$  needs  $x$  and  $w$  only to compute the last round and  $\mathcal{V}$  needs  $x$  only to compute the output. Before that,  $\mathcal{P}$  and  $\mathcal{V}$  run having as input only the size of  $x$ . The notion of delayed-input completeness was defined in [8]. An interactive protocol  $\Pi = (\mathcal{P}, \mathcal{V})$  is *public coin* if, at every round,  $\mathcal{V}$  simply tosses a predetermined number of coins (random challenge) and sends the outcome to the prover. We say that the transcript  $\tau$  of an execution  $b = \langle \mathcal{P}(z), \mathcal{V} \rangle(x)$  is *accepting* if  $b = 1$ .

**Definition 2 (Special Honest-Verifier Zero Knowledge (Special HVZK)).**

Consider a public-coin proof/argument system  $\Pi = (\mathcal{P}, \mathcal{V})$  for an  $\mathcal{NP}$ -language  $L$  where the verifier sends  $m$  messages of length  $\ell_1, \dots, \ell_m$ . We say that  $\Pi$  is Special HVZK if there exists a PPT simulator algorithm  $\mathcal{S}$  that on input any  $x \in L$ , security parameter  $1^\lambda$  and any  $c_1 \in \{0, 1\}^{\ell_1}, \dots, c_m \in \{0, 1\}^{\ell_m}$ , outputs a transcript for proving  $x \in L$  where  $c_1, \dots, c_m$  are the messages of the verifier, such that the distribution of the output of  $\mathcal{S}$  is computationally indistinguishable from the distribution of a transcript obtained when  $\mathcal{V}$  sends  $c_1, \dots, c_m$  as challenges and  $\mathcal{P}$  runs on common input  $x$  and any  $w$  such that  $(x, w) \in \text{Rel}_L$ .

In this paper we consider the notion of proof/argument of knowledge (PoK/AoK) defined in [29]. Furthermore we consider the *adaptive-input* PoK/AoK property for all the protocols that enjoy delayed-input completeness. Adaptive-input PoK/AoK ensures that the PoK/AoK property still holds when a malicious prover can choose the statement adaptively at the last round. We consider the 3-round public-coin Special HVZK PoK proposed by Lapidot and Shamir [28], that we denote by LS. LS enjoys delayed-input completeness since the inputs for both  $\mathcal{P}$  and  $\mathcal{V}$  are needed only to play the last round, and only the length of the instance is needed earlier. LS also enjoys adaptive-input PoK. In particular we use a 4-round delayed-input special HVZK adaptive-input AoK that is a variant of LS [13] that relies on OWFs only. The additional round is indeed needed to instantiate the commitment scheme used in LS under any OWF.

## 2.2 2-Round Instance-Dependent Trapdoor Commitments

Here we define a special commitment scheme based on an  $\mathcal{NP}$ -language  $L$  where sender and receiver also receive as input an instance  $x$ . While correctness and computational hiding hold for any  $x$ , we require that statistical binding holds

for  $x \notin L$  and knowledge of a witness for  $x \in L$  allows to equivocate. Finally, we require that a commitment along with two different openings allows to compute the witness for  $x \in L$ . We recall that  $\hat{L}$  denotes the language that includes  $L$  and all well formed instances that are not in  $L$ .

**Definition 3.** Let  $1^\lambda$  be the security parameter,  $L$  be an  $\mathcal{NP}$ -language and  $\text{Rel}_L$  be the corresponding  $\mathcal{NP}$ -relation. A triple of PPT algorithms  $\text{TC} = (\text{Sen}, \text{Rec}, \text{TFake})$  is a 2-Round Instance-Dependent Trapdoor Commitment scheme if the following properties hold.

**Correctness.** In the 1st round,  $\text{Rec}$  on input  $1^\lambda$  and  $x \in \hat{L}$  outputs  $\rho$ . In the 2nd round  $\text{Sen}$  on input the message  $m$ ,  $1^\lambda$ ,  $\rho$  and  $x \in L$  outputs  $(\text{com}, \text{dec})$ . We will refer to the pair  $(\rho, \text{com})$  as the commitment of  $m$ . Moreover we will refer to the execution of the above two rounds including the exchange of the corresponding two messages as the commitment phase. Then  $\text{Rec}$  on input  $m$ ,  $x$ ,  $\text{com}$ ,  $\text{dec}$  and the private coins used to generate  $\rho$  in the commitment phase outputs 1. We will refer to the execution of this last round including the exchange of  $\text{dec}$  as the decommitment phase. Notice that an adversarial sender  $\text{Sen}^*$  could deviate from the behavior of  $\text{Sen}$  when computing and sending  $\text{com}$  and  $\text{dec}$  for an instance  $x \in \hat{L}$ . As a consequence  $\text{Rec}$  could output 0 in the decommitment phase. We will say that  $\text{dec}$  is a valid decommitment of  $(\rho, \text{com})$  to  $m$  for an instance  $x \in \hat{L}$ , if  $\text{Rec}$  outputs 1.

**Hiding.** Given a PPT adversary  $\mathcal{A}$ , consider the following hiding experiment  $\text{ExpHiding}_{\mathcal{A}, \text{TC}}^b(\lambda, x)$  for  $b = 0, 1$  and  $x \in \hat{L}_R$ :

- On input  $1^\lambda$  and  $x$ ,  $\mathcal{A}$  outputs a message  $m$ , along with  $\rho$ .
  - The challenger on input  $x, m, \rho, b$  works as follows: if  $b = 0$  then it runs  $\text{Sen}$  on input  $m, x$  and  $\rho$ , obtaining a pair  $(\text{com}, \text{dec})$ , otherwise it runs  $\text{TFake}$  on input  $x$  and  $\rho$ , obtaining a pair  $(\text{com}, \text{aux})$ . The challenger outputs  $\text{com}$ .
  - $\mathcal{A}$  on input  $\text{com}$  outputs a bit  $b'$  and this is the output of the experiment.
- We say that hiding holds if for any PPT adversary  $\mathcal{A}$  there exist a negligible function  $\nu$ , s.t.:

$$\left| \text{Prob} [ \text{ExpHiding}_{\mathcal{A}, \text{TC}}^0(\lambda, x) = 1 ] - \text{Prob} [ \text{ExpHiding}_{\mathcal{A}, \text{TC}}^1(\lambda, x) = 1 ] \right| < \nu(\lambda).$$

**Special Binding.** There exists a PPT algorithm that on input a commitment  $(\rho, \text{com})$ , the private coins used by  $\text{Rec}$  to compute  $\rho$ , and two valid decommitments  $(\text{dec}, \text{dec}')$  of  $(\rho, \text{com})$  to two different messages  $m$  and  $m'$  w.r.t. an instance  $x \in L$ , outputs  $w$  s.t.  $(x, w) \in \text{Rel}_L$  with overwhelming probability.

**Trapdooriness.** For any PPT adversary  $\mathcal{A}$  there exist a negligible function  $\nu$ , s.t. for all  $x \in L$  it holds that:

$$\left| \text{Prob} [ \text{ExpCom}_{\mathcal{A}, \text{TC}}(\lambda, x) = 1 ] - \text{Prob} [ \text{ExpTrapdoor}_{\mathcal{A}, \text{TC}}(\lambda, x) = 1 ] \right| < \nu(\lambda)$$

where  $\text{ExpCom}_{\mathcal{A}, \text{TC}}(\lambda, x)$  and  $\text{ExpTrapdoor}_{\mathcal{A}, \text{TC}}(\lambda, x)$  are defined below<sup>17</sup>.

<sup>17</sup> We assume w.l.o.g. that  $\mathcal{A}$  is stateful.

$\text{ExpCom}_{\mathcal{A}, \text{TC}}(\lambda, x):$ -On input $1^\lambda$ and $x$ , $\mathcal{A}$ outputs $(\rho, m)$ . -Sen on input $1^\lambda$ , $x$ , $m$ and $\rho$ , outputs $(\text{com}, \text{dec})$ .  - $\mathcal{A}$ on input $(\text{com}, \text{dec})$ outputs a bit $b$ and this is the output of the experiment.	$\text{ExpTrapdoor}_{\mathcal{A}, \text{TC}}(\lambda, x):$ -On input $1^\lambda$ and $x$ , $\mathcal{A}$ outputs $(\rho, m)$ . -TFake on input $1^\lambda$ , $x$ and $\rho$ , outputs $(\text{com}, \text{aux})$ . -TFake on input $\text{tk}$ s.t. $(x, \text{tk}) \in \text{Rel}_L$ , $x$ , $\rho$ , $\text{com}$ , $\text{aux}$ and $m$ outputs $\text{dec}$ .  - $\mathcal{A}$ on input $(\text{com}, \text{dec})$ outputs a bit $b$ and this is the output of the experiment.
--	---

### 2.3 Non-Malleable Commitments

Here we follow [31]. Let  $\Pi = (\text{Sen}, \text{Rec})$  be a statistically binding commitment scheme and let  $\lambda$  be the security parameter. Consider MiM adversaries that are participating in left and right sessions in which  $\text{poly}(\lambda)$  commitments take place. We compare between a MiM and a simulated execution. In the MiM execution the adversary  $\mathcal{A}$ , with auxiliary information  $z$ , is simultaneously participating in  $\text{poly}(\lambda)$  left and right sessions. In the left sessions the MiM adversary  $\mathcal{A}$  interacts with  $\text{Sen}_1, \dots, \text{Sen}_{\text{poly}(\lambda)}$  receiving commitments to values  $m_1, \dots, m_{\text{poly}(\lambda)}$  using identities  $\text{id}_1, \dots, \text{id}_{\text{poly}(\lambda)}$  of its choice. In the right session  $\mathcal{A}$  interacts with  $\text{Rec}_1, \dots, \text{Rec}_{\text{poly}(\lambda)}$  attempting to commit to a sequence of related values  $\tilde{m}_1, \dots, \tilde{m}_{\text{poly}(\lambda)}$  again using identities of its choice  $\tilde{\text{id}}_1, \dots, \tilde{\text{id}}_{\text{poly}(\lambda)}$ . If any of the right commitments is invalid, or undefined, its value is set to  $\perp$ . For any  $i$  such that  $\tilde{\text{id}}_i = \text{id}_j$  for some  $j$ , set  $\tilde{m}_i = \perp$  (i.e., any commitment where the adversary uses the same identity of one of the honest senders is considered invalid). Let  $\text{mim}_{\Pi}^{\mathcal{A}, m_1, \dots, m_{\text{poly}(\lambda)}}(z)$  denote a random variable that describes the values  $\tilde{m}_1, \dots, \tilde{m}_{\text{poly}(\lambda)}$  and the view of  $\mathcal{A}$ , in the above experiment. In the simulated execution, an efficient simulator  $S$  directly interacts with  $\text{Rec}_1, \dots, \text{Rec}_{\text{poly}(\lambda)}$ . Let  $\text{sim}_{\Pi}^S(1^\lambda, z)$  denote the random variable describing the values  $\tilde{m}_1, \dots, \tilde{m}_{\text{poly}(\lambda)}$  committed by  $S$ , and the output view of  $S$ ; whenever the view contains in the  $i$ -th right session the same identity of any of the identities of the left sessions, then  $m_i$  is set to  $\perp$ .

In all the paper we denote by  $\tilde{\delta}$  a value associated with the right session (where the adversary  $\mathcal{A}$  plays with a receiver) where  $\delta$  is the corresponding value in the left session. For example, the sender commits to  $v$  in the left session while  $\mathcal{A}$  commits to  $\tilde{v}$  in the right session.

**Definition 4 (Concurrent NM commitment scheme [31]).** *A commitment scheme is concurrent NM with respect to commitment (or a many-many NM commitment scheme) if, for every PPT concurrent MiM adversary  $\mathcal{A}$ , there exists a PPT simulator  $S$  such that for all  $m_i \in \{0, 1\}^{\text{poly}(\lambda)}$  for  $i = 1, \dots, \text{poly}(\lambda)$  the following ensembles are computationally indistinguishable:*

$$\{\text{mim}_{\Pi}^{\mathcal{A}, m_1, \dots, m_{\text{poly}(\lambda)}}(z)\}_{z \in \{0, 1\}^*} \approx \{\text{sim}_{\Pi}^S(1^\lambda, z)\}_{z \in \{0, 1\}^*}.$$

As in [31] we also consider relaxed notions of concurrent non-malleability: one-many and one-one NM commitment schemes. In a one-many NM commitment scheme,  $\mathcal{A}$  participates in one left and polynomially many right sessions. In a one-one (i.e., a stand-alone secure) NM commitment scheme, we consider only adversaries  $\mathcal{A}$  that participate in one left and one right session. We will make use of the following proposition of [31].

**Proposition 1.** *Let (Sen, Rec) be a one-many NM commitment scheme. Then, (Sen, Rec) is also a concurrent (i.e., many-many) NM commitment scheme.*

We say that a commitment is valid or well formed if it can be decommitted to a message  $m \neq \perp$ . Following [29] we say that a MiM is *synchronous* if it “aligns” the left and the right sessions; that is, whenever it receives message  $i$  on the left, it directly sends message  $i$  on the right, and vice versa.

## 2.4 New Definitions: weak NM and SimWI

**Definition 5 (weak NM commitment scheme).** *A commitment scheme is weak one-one (resp., one-many) non-malleable if it is a one-one (resp., one-many) NM commitment scheme with respect to MiM adversary that when receiving a well formed commitment in the left session, except with negligible probability computes well formed commitments (i.e., the computed commitments can be opened to messages  $\neq \perp$ ) in the right sessions.*

In the rest of the paper, following [25], we assume that identities are known before the protocol begins, though strictly speaking this is not necessary, as the identities do not appear in the protocol until after the first committer message. The MiM can choose his identity adversarially as long as it differs from the identities used by honest senders. As already observed in previous work, when the identity is selected by the sender the id-based definitions guarantee non-malleability as long as the MiM does not behave like a proxy (an unavoidable attack). Indeed the sender can pick as identity the public key of a signature scheme signing the transcript. The MiM will have to use a different identity or to break the signature scheme.

*Simulation-witness-independence (SimWI) for  $L^\gamma$ .* We define SimWI for an  $\mathcal{NP}$  language  $L$  associating to the language a non-negative integer  $\gamma$ . Roughly speaking all witnesses of an instance have in common the first  $\gamma$  bits, and this property holds for all instances of  $L$ . More formally we will consider  $\gamma$  as a non-negative integer such that for any  $x \in L$  it holds that any witness  $w$  of  $x$  can be parsed as  $w = \alpha|\beta$ , where  $|\alpha| = \gamma$ , and  $\alpha$  is the same for all witnesses of  $x$ . In order to ease the notation, we will note denote by  $L^\gamma$  the  $\mathcal{NP}$  language having the above prefix  $\gamma$ . We will say that  $L^\gamma$  is a  $\gamma$ -prefix language meaning that for any instance  $x$  of  $L^\gamma$  all witnesses of  $x$  have the same first  $\gamma$  bits.

When defining SimWI we will consider the one-many case since this is what we will use in the next part of the paper. Adapting the definition to the one-one case and to the fully concurrent case is straightforward.

**Discussion on adaptive-input selection and black-box simulation.**

Since our definition considers a real game where the MiM plays with at most one prover and polynomially many verifiers, and a simulated game that consists of an execution of a stand-alone simulator, a natural definition would require the indistinguishability of the two games for any  $x \in L^\gamma$ , giving to the prover as input also a witness. This definition however would be difficult to use when the argument of knowledge is played as a subprotocol of a larger protocol, especially if it is played in parallel with other subprotocols and the adversary contributes in selecting the statement for the left session. More specifically applications require a security definition that features a delayed-input property so that players start the protocol with the common input that is still undefined, and that will be defined later potentially with the contribution of the adversary. Therefore in our definition we will allow the adversary to explicitly select the statement, and as such the adversary will provide also the witness for the prover. The simulated game however will filter out the witness so that the simulator will receive only the instance. This approach strictly follows the one of [50] where adaptive-input selection is explicitly allowed and managed in a similar way. As final remark, our definition will require the existence of a black-box simulator since a non-black-box simulator could retrieve from the code of the adversary the witness for the adaptively generated statement. The non-black-box simulator could then run the honest prover procedure, therefore canceling completely the security flavor of the simulation paradigm.

For simplicity we now give the formal definition with non-delayed inputs.

**Definition.** Let  $\Pi = (\mathcal{P}, \mathcal{V})$  be an argument system for a  $\gamma$ -prefix language  $L^\gamma$  and let  $\text{Rel}_{L^\gamma}$  be the corresponding witness relation. Consider a PPT MiM adversary  $\mathcal{A}$  that is simultaneously participating in one left session and  $\text{poly}(\lambda)$  right sessions. When the execution starts, all parties receive as a common input the security parameter  $1^\lambda$  then  $\mathcal{A}$  chooses the statement  $x \in L^\gamma$  and witness  $w$  s.t.  $(x, w) \in \text{Rel}_{L^\gamma}$  and sends them to  $\mathcal{P}$ , furthermore  $\mathcal{A}$  receives as auxiliary input  $z \in \{0, 1\}^*$ .

In the left session an honest prover  $\mathcal{P}$  interacting with  $\mathcal{A}$  proves the membership of  $x$  in  $L^\gamma$ . In the  $\text{poly}(\lambda)$  right sessions,  $\mathcal{A}$  proves the membership in  $L^\gamma$  of instances  $\tilde{x}_1, \dots, \tilde{x}_{\text{poly}(\lambda)}$  of his choice to the honest verifiers  $\mathcal{V}_1, \dots, \mathcal{V}_{\text{poly}(\lambda)}$ . For simplicity, in this definition we consider an adversary  $\mathcal{A}$  that chooses the statement to be proved in the 1st round that he plays in every right sessions<sup>18</sup>.

Let  $\{\text{wimim}_\Pi(1^\lambda, z)\}_{\lambda \in \mathbb{N}, z \in \{0, 1\}^*}$  be a random variable that describes the following 3 values: 1) the view of  $\mathcal{A}$  in the above experiment, 2) the output of  $\mathcal{V}_i$  for  $i = 1, \dots, \text{poly}(\lambda)$  and 3) the first  $\gamma$  bits  $\tilde{w}_1^\gamma, \dots, \tilde{w}_{\text{poly}(\lambda)}^\gamma$  of the corresponding witnesses  $\tilde{w}_1, \dots, \tilde{w}_{\text{poly}(\lambda)}$  w.r.t. the instances  $\tilde{x}_1, \dots, \tilde{x}_{\text{poly}(\lambda)}$  that are part of  $\mathcal{A}$ 's view except that  $\tilde{w}_i^\gamma = \perp$  if  $\mathcal{V}_i$  did not output 1, with  $i = 1, \dots, \text{poly}(\lambda)$ .

<sup>18</sup> Our construction will satisfy a much stronger notion where in the left session  $\mathcal{A}$  can choose statement and witness in the last round, while in the right sessions  $\mathcal{A}$  can choose the statement in the very last round, as long as the witness is already fixed in the second round.



Let  $\{\text{sim}_\Pi^S(1^\lambda, z)\}_{\lambda \in \mathbb{N}, z \in \{0,1\}^*}$  be a random variable that describes the following 3 values: 1) and 2) correspond to the output of  $\mathcal{S}$ , 3) consists of the first  $\gamma$  bits  $\tilde{w}_1^\gamma, \dots, \tilde{w}_{\text{poly}(\lambda)}^\gamma$  of the corresponding witnesses  $\tilde{w}_1, \dots, \tilde{w}_{\text{poly}(\lambda)}$  w.r.t. the instances  $\tilde{x}_1, \dots, \tilde{x}_{\text{poly}(\lambda)}$  that appear in the MiM view of the output of  $\mathcal{S}$  except that  $w_i^\gamma = \perp$  if  $b_i = 0$ . The output of  $\mathcal{S}$  is composed by the following two values: 1) a MiM view and 2) bits  $b_1, \dots, b_{\text{poly}(\lambda)}$ .

$\mathcal{S}$  has black-box access to  $\mathcal{A}$  and has the goal to emulate the prover without having a witness, while perfectly emulating the verifiers of the right sessions. Therefore  $\mathcal{S}$  rewinds only when playing as prover<sup>19</sup> and every instance/witness pair  $(x, w)$  given in output by  $\mathcal{A}$  is replaced by  $(x, \perp)$  and then returned to  $\mathcal{S}$  (i.e., the simulator runs without the witness  $w$  for the instance  $x$  chosen by  $\mathcal{A}$ ).

**Definition 6 (SimWI).** *An argument system  $\Pi = (\mathcal{P}, \mathcal{V})$  for a  $\gamma$ -prefix language  $L^\gamma$  with witness relation  $\text{Rel}_{L^\gamma}$  is SimWI if there exists an expected polynomial-time simulator  $\mathcal{S}$  such that for every MiM adversary  $\mathcal{A}$  that participates in one left session and  $\text{poly}(\lambda)$  right sessions the ensembles  $\{\text{wimim}_\Pi(1^\lambda, z)\}_{\lambda \in \mathbb{N}, z \in \{0,1\}^*}$  and  $\{\text{sim}_\Pi^S(1^\lambda, z)\}_{\lambda \in \mathbb{N}, z \in \{0,1\}^*}$  are computationally indistinguishable over  $\lambda$ .*

### 3 4-Round One-Many SimWI From OWFs

We now show our construction of a 4-round argument of knowledge SWI =  $(\mathcal{P}^{\text{swi}}, \mathcal{V}^{\text{swi}})$  for the  $\gamma$ -prefix language  $L^\gamma$  that is one-many SimWI and can be instantiated using any OWF. We will need the following tools:

1. a signature scheme  $\Sigma = (\text{Gen}, \text{Sign}, \text{Ver})$ ;
2. a 2-round IDTC scheme  $\text{TC}_\Sigma = (\text{Sen}_\Sigma, \text{Rec}_\Sigma, \text{TFake}_\Sigma)$  for the following  $\mathcal{NP}$ -language

$$L_\Sigma = \left\{ \text{vk} : \exists (\text{msg}_1, \text{msg}_2, \sigma_1, \sigma_2) \text{ s.t. } \text{Ver}(\text{vk}, \text{msg}_1, \sigma_1) = 1 \right. \\ \left. \text{AND } \text{Ver}(\text{vk}, \text{msg}_2, \sigma_2) = 1 \text{ AND } \text{msg}_1 \neq \text{msg}_2 \right\};$$

3. a 4-round delayed-input public-coin Special HVZK (Def. 2) proof system  $\text{LS} = (\mathcal{P}, \mathcal{V})$  for the  $\gamma$ -prefix language  $L^\gamma$  that is adaptive-input PoK for the corresponding relation  $\text{Rel}_{L^\gamma}$ .

<sup>19</sup> The motivation behind this definitional choice is that  $\mathcal{S}$  is supposed to be an extended zero-knowledge simulator that takes care also of the honest behavior of the verifiers since  $\mathcal{A}$  expects to play with them. Instead allowing  $\mathcal{S}$  to have any behavior on the right would hurt the power of SimWI in composing in parallel with other protocols. Indeed if  $\mathcal{S}$  rewinds on the right as verifier, it would in turn rewind also the left player of the external protocol that is played in parallel. This would hurt the security of the overall scheme whenever the external protocol is not resettably secure. We will indeed compose a SimWI AoK with a weak NM commitment scheme that is not resettably secure.

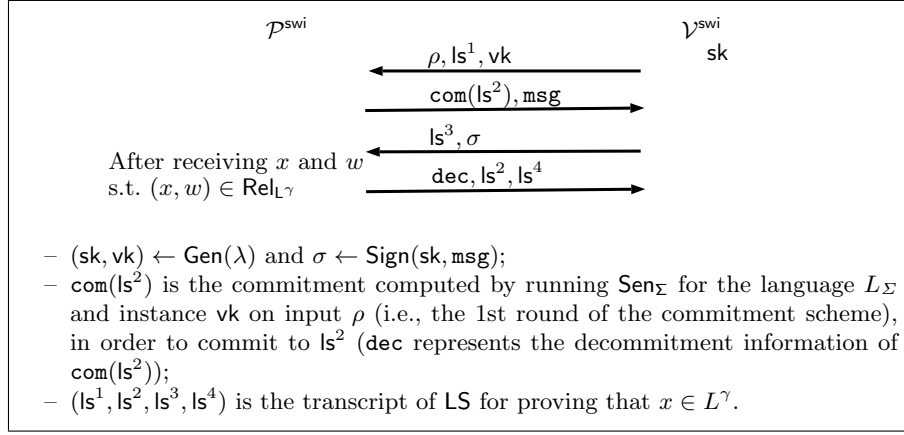


Fig. 1: 4-Round SimWI AoK SWI from OWFs.

Let  $x \in L^\gamma$  be the statement that  $\mathcal{P}^{\text{swi}}$  wants to prove, and  $w$  a witness s.t.  $(x, w) \in \text{Rel}_{L^\gamma}$ . The high-level idea of our protocol is depicted in Fig. 1. In the 1st round the verifier  $\mathcal{V}^{\text{swi}}$  computes and sends the 1st round  $\text{ls}^1$  of LS, computes a pair of signature and verification keys  $(\text{sk}, \text{vk})$ , sends the verification key  $\text{vk}$  to  $\mathcal{P}^{\text{swi}}$  and computes and sends the 1st round  $\rho$  of  $\text{TC}_\Sigma$  by running  $\text{Rec}_\Sigma$  on input  $1^\lambda$  and the instance  $\text{vk} \in L_\Sigma$ . Then  $\mathcal{P}^{\text{swi}}$  on input  $x, w$  and the received 1st round, computes the 2nd round  $\text{ls}^2$  of LS and runs  $\text{Sen}_\Sigma$  on input  $1^\lambda, \text{vk}, \rho$  and message  $\text{ls}^2$  thus obtaining a pair  $(\text{com}, \text{dec})$ .  $\mathcal{P}^{\text{swi}}$  sends  $\text{com}$  and a random message  $\text{msg}$  to  $\mathcal{V}^{\text{swi}}$ . In the 3rd round  $\mathcal{V}^{\text{swi}}$  sends the 3rd round  $\text{ls}^3$  of LS and a signature  $\sigma$  (computed using  $\text{sk}$ ) of the message  $\text{msg}$ . In the last round  $\mathcal{P}^{\text{swi}}$  verifies whether or not  $\sigma$  is a valid signature for  $\text{msg}$ . If  $\sigma$  is a valid signature, then  $\mathcal{P}^{\text{swi}}$ , using  $x, w$  and  $\text{ls}^3$ , computes the 4th round  $\text{ls}^4$  of LS and sends  $\text{dec}, \text{ls}^2$  and  $\text{ls}^4$  to  $\mathcal{V}^{\text{swi}}$ . At this point  $\mathcal{V}^{\text{swi}}$  outputs 1 iff  $\text{Rec}_\Sigma$  on input  $\text{vk}, \text{com}, \text{dec}, \text{ls}^2$  accepts  $(\text{ls}^2, \text{dec})$  as a decommitment of  $\text{com}$  and the transcript for LS is accepting for  $\mathcal{V}$  with respect to the instance  $x$ . We remark that to execute LS the instance is not needed until the last round but the instance length is required from the onset of the protocol.

Fig. 2 describes in details our SimWI AoK SWI.

**Theorem 1.** *Assuming OWFs,  $\text{SWI} = (\mathcal{P}^{\text{swi}}, \mathcal{V}^{\text{swi}})$  is a 4-round one-many SimWI AoK for  $\gamma$ -prefix languages.*

We divide the security proof in three parts, proving that SWI enjoys delayed-input completeness, adaptive-input AoK and SimWI. Before that, we recall that LS can be constructed from OWFs (see Section 2.1) as well as  $\Sigma$  using [48]. We also observe that if  $\Sigma$  relies on OWFs, then also  $\text{TC}_\Sigma$  can be constructed from OWFs (see the full version [5]).

**Delayed-Input Completeness.** The completeness follows directly from the completeness of LS, the correctness of  $\text{TC}_\Sigma$  and the validity of  $\Sigma$ . We observe that, due to the delayed-input property of LS, the statement  $x$  (and the respective

- Common input:** security parameter  $\lambda$ , instance  $x \in L^\gamma$ , instance length  $\ell$ .  
**Input to  $\mathcal{P}^{\text{swi}}$ :**  $w$  s.t.  $(x, w) \in \text{Rel}_{L^\gamma}$ , with  $x, w$  available only in the 4th round.  
**Commitment phase:**
1.  $\mathcal{V}^{\text{swi}} \rightarrow \mathcal{P}^{\text{swi}}$ 
    - 1.1. Run  $(\text{sk}, \text{vk}) \leftarrow \text{Gen}(1^\lambda)$ .
    - 1.2. Run  $\mathcal{V}$  on input  $1^\lambda$  and  $\ell$  thus obtaining the 1st round  $\text{ls}^1$  of LS.
    - 1.3. Run  $\text{Rec}_\Sigma$  on input  $1^\lambda$  and  $\text{vk}$  thus obtaining  $\rho$ .
    - 1.4. Send  $(\text{vk}, \text{ls}^1, \rho)$  to  $\mathcal{P}^{\text{swi}}$ .
  2.  $\mathcal{P}^{\text{swi}} \rightarrow \mathcal{V}^{\text{swi}}$ 
    - 2.1. Run  $\mathcal{P}$  on input  $1^\lambda$ ,  $\ell$  and  $\text{ls}^1$  thus obtaining the 2nd round  $\text{ls}^2$  of LS.
    - 2.2. Run  $\text{Sen}_\Sigma$  on input  $1^\lambda$ ,  $\text{vk}$ ,  $\rho$  and message  $\text{ls}^2$  to compute the pair  $(\text{com}, \text{dec})$ .
    - 2.3. Pick a message  $\text{msg} \leftarrow \{0, 1\}^\lambda$ .
    - 2.4. Send  $(\text{com}, \text{msg})$  to  $\mathcal{V}^{\text{swi}}$ .
  3.  $\mathcal{V}^{\text{swi}} \rightarrow \mathcal{P}^{\text{swi}}$ 
    - 3.1. Run  $\mathcal{V}$  thus obtaining the 3rd round  $\text{ls}^3$  of LS.
    - 3.2. Run  $\text{Sign}(\text{sk}, \text{msg})$  thus obtaining a signature  $\sigma$  of the message  $\text{msg}$ .
    - 3.3. Send  $(\text{ls}^3, \sigma)$  to  $\mathcal{P}^{\text{swi}}$ .
  4.  $\mathcal{P}^{\text{swi}} \rightarrow \mathcal{V}^{\text{swi}}$ 
    - 4.1. If  $\text{Ver}(\text{vk}, \text{msg}, \sigma) \neq 1$  then abort, continue as follows otherwise.
    - 4.2. Run  $\mathcal{P}$  on input  $x$ ,  $w$  and  $\text{ls}^3$  thus obtaining the 4th round  $\text{ls}^4$  of LS.
    - 4.3. Send  $((\text{dec}, \text{ls}^2), \text{ls}^4)$  to  $\mathcal{V}^{\text{swi}}$ .
  5.  $\mathcal{V}^{\text{swi}}$ : output 1 iff the following conditions are satisfied.
    - 5.1.  $\text{Rec}_\Sigma$  on input  $\text{vk}, \text{com}, \text{dec}, \text{ls}^2$  accepts  $(\text{ls}^2, \text{dec})$  as a decommitment of  $\text{com}$ .
    - 5.2.  $(\text{ls}^1, \text{ls}^2, \text{ls}^3, \text{ls}^4)$  is accepting for  $\mathcal{V}$  with respect to the instance  $x$ .

Fig. 2: 4-Round SimWI AoK SWI from OWFs.

witness  $w$ ) are used by  $\mathcal{P}^{\text{swi}}$  only to compute the last round. Therefore SWI enjoys delayed-input completeness.

**Adaptive-Input Argument of Knowledge.** In order to prove that SWI enjoys adaptive-input AoK for  $\text{Rel}_{L^\gamma}$ , we need to show an efficient extractor  $\text{E}$  that outputs the witnesses for the statements proved by an adversarial prover  $\mathcal{P}^{\text{swi}^*}$ .  $\text{E}$  simply runs  $\text{ExtLS}$ , the adaptive-input PoK extractor of LS, in every right session, and outputs what  $\text{ExtLS}$  outputs. More precisely  $\text{E}$  internally runs and interacts with a SWI prover  $\mathcal{P}^{\text{swi}}$  as  $\mathcal{V}_i^{\text{swi}}$  does, but acting as a proxy between  $\mathcal{P}^{\text{swi}^*}$  and  $\text{ExtLS}$  w.r.t. the messages of LS (for  $i = 1, \dots, \text{poly}(\lambda)$ ). The important observation is that  $\text{E}$  could fail if the following event  $\text{NoExt}$  happens with non-negligible probability:  $\mathcal{P}^{\text{swi}^*}$  opens the commitment  $(\rho, \text{com})$  to a different  $\text{ls}^2$  during the rewinds. Indeed, in this case  $\text{ExtLS}$  could fail in obtaining a witness. We prove the following claim.

**Claim 1.** There exists a negligible function  $\nu$  such that  $\text{Prob}[\text{NoExt}] < \nu(\lambda)$ .

*Proof.* The proof is by contradiction, more specifically we now show an adversary  $\mathcal{A}^\Sigma$  that extracts two signatures for two different messages in order to break the signature scheme  $\Sigma$  when  $\text{Prob}[\text{NoExt}]$  is non-negligible in  $\lambda$ .

If two decommitments of  $(\text{com}, \rho)$  w.r.t. two different messages ( $\text{ls}^{2'}$  and  $\text{ls}^2$ ) are shown by  $\mathcal{P}^{\text{swi}^*}$  in the last round of SWI,  $\mathcal{A}^\Sigma$  can extract two different signatures for two different messages by using the special binding of  $\text{TC}_\Sigma$ . More precisely, let  $\text{vk}$  be the verification key given by the challenger of the signature scheme, then our adversary  $\mathcal{A}^\Sigma$  works as follows.

For all  $i \in \{1, \dots, \text{poly}(\lambda)\} - \{j\}$ ,  $\mathcal{A}^\Sigma$  interacts in the  $i$ -th session against  $\mathcal{P}^{\text{swi}^*}$  as  $\mathcal{V}_i^{\text{swi}}$  would do. Instead in the  $j$ -th session  $\mathcal{A}^\Sigma$  runs as  $\text{E}$  would do, using  $\text{vk}$  to compute the first round, and the oracle  $\text{Sign}(\text{sk}, \cdot)$  to compute a signature  $\sigma$  of a message  $m$  sent by  $\mathcal{A}^\Sigma$  in the second round. Since we are assuming (by contradiction) that during the rewinds from the 4th round to the 3rd round the commitment  $(\rho, \text{com})$  (sent in the second round by  $\mathcal{P}^{\text{swi}^*}$ ) is opened in more than one way, then, by using the special binding of  $\text{TC}_\Sigma$ ,  $\mathcal{A}^\Sigma$  extracts and outputs two signatures for two different messages. We conclude this proof with the following two observations. First, the signature oracle  $\text{Sign}(\text{sk}, \cdot)$  is called only once since, by construction of  $\text{E}$ , the second round is played by  $\mathcal{P}^{\text{swi}^*}$  only once. Second, the extractor  $\text{E}$  is an expected polynomial-time algorithm while  $\mathcal{A}^\Sigma$  must be a strict polynomial-time algorithm. This mean that the execution  $\text{E}$  has to be truncated. Obviously the running time of the extraction procedure can be truncated to a sufficiently long value so that with non-negligible probability the truncated extraction procedure will still yield the event  $\text{NoExt}$  to happened and this is sufficient for  $\mathcal{A}^\Sigma$  to break the signature scheme<sup>20</sup>.

**SimWI.** In order to prove that SWI is SimWI (Definition 6) for any  $\gamma$ -prefix language  $L^\gamma$  we prove the following lemma.

**Lemma 1.**  $\{\text{wimim}_{\text{SWI}}(1^\lambda, z)\}_{\lambda \in \mathbb{N}, z \in \{0,1\}^*} \approx \{\text{sim}_{\text{SWI}}^{\text{swi}}(1^\lambda, z)\}_{\lambda \in \mathbb{N}, z \in \{0,1\}^*}$ .

*Proof.* Here we actually prove something stronger. Indeed we prove the security of SWI considering a MiM adversary  $\mathcal{A}^{\text{swi}}$  that has additional power both in the left and in the right sessions. More precisely in the left session  $\mathcal{A}^{\text{swi}}$  can choose the statement to be proved (and the related witness) in the third round. That is, in the last round that goes from  $\mathcal{A}^{\text{swi}}$  to  $\mathcal{P}^{\text{swi}}$ .

Also, in all right sessions  $\mathcal{A}$  fixes a family of statements in the second round, and then adaptively picks the statement to be proved from that family in the last round. In this way the MiM adversary has the power to adaptively choose the statement to be proved in the last round of every right session conditioned on belonging to the already fixed family, that has to be fixed in the second round. In the rest of the paper we will refer to a SimWI protocol that is secure also in this setting as *adaptive-input* SimWI.

<sup>20</sup> The same arguments are used in [17]. The same standard argument about truncating the execution of an expected polynomial-time algorithm is used in another proofs but for simplicity we will not repeat this discussion.

We start by showing the simulator  $\mathcal{S}^{\text{swi}}$  and giving an overview of the entire proof. The simulator is described in Figure 3. Roughly,  $\mathcal{S}^{\text{swi}}$  interacts against  $\mathcal{A}^{\text{swi}}$  in both the left and right sessions. In the left session  $\mathcal{S}^{\text{swi}}$  runs  $\text{TFake}_\Sigma$  to compute and send a commitment  $\text{com}$ .  $\mathcal{S}^{\text{swi}}$  then rewinds  $\mathcal{A}^{\text{swi}}$  from the 3rd to the 2nd round, in order to obtain two valid signatures  $\sigma_1, \sigma_2$  for two different messages  $(\text{msg}_1, \text{msg}_2)$ . This information constitutes the trapdoor  $\text{tk}$  for  $\text{TC}_\Sigma$ . After that  $\text{tk}$  is computed,  $\mathcal{S}^{\text{swi}}$  comes back to the main thread execution. Upon receiving  $\text{ls}^3$  and  $x$  in the 3rd round from  $\mathcal{A}^{\text{swi}}$ ,  $\mathcal{S}^{\text{swi}}$  computes an accepting transcript for LS  $(\text{ls}^1, \text{ls}^2, \text{ls}^3, \text{ls}^4)$  running the Special HVZK simulator of LS on input  $\text{ls}^1$ , received in the 1st round from  $\mathcal{A}^{\text{swi}}$ , and  $(x, \text{ls}^3)$ . In the last round computes, by using  $\text{tk}$ , the decommitment information  $(\text{dec}, \text{ls}^2)$  for  $\text{com}$ , and sends  $(\text{dec}, \text{ls}^2, \text{ls}^4)$  to  $\mathcal{A}^{\text{swi}}$ . In the  $i$ -th right session, for  $i = 1, \dots, \text{poly}(\lambda)$ ,  $\mathcal{S}^{\text{swi}}$  acts as  $\mathcal{V}_i^{\text{swi}}$  would do against  $\mathcal{A}^{\text{swi}}$ . When the execution against  $\mathcal{A}^{\text{swi}}$  ends,  $\mathcal{S}^{\text{swi}}$  outputs the view of  $\mathcal{A}^{\text{swi}}$ .

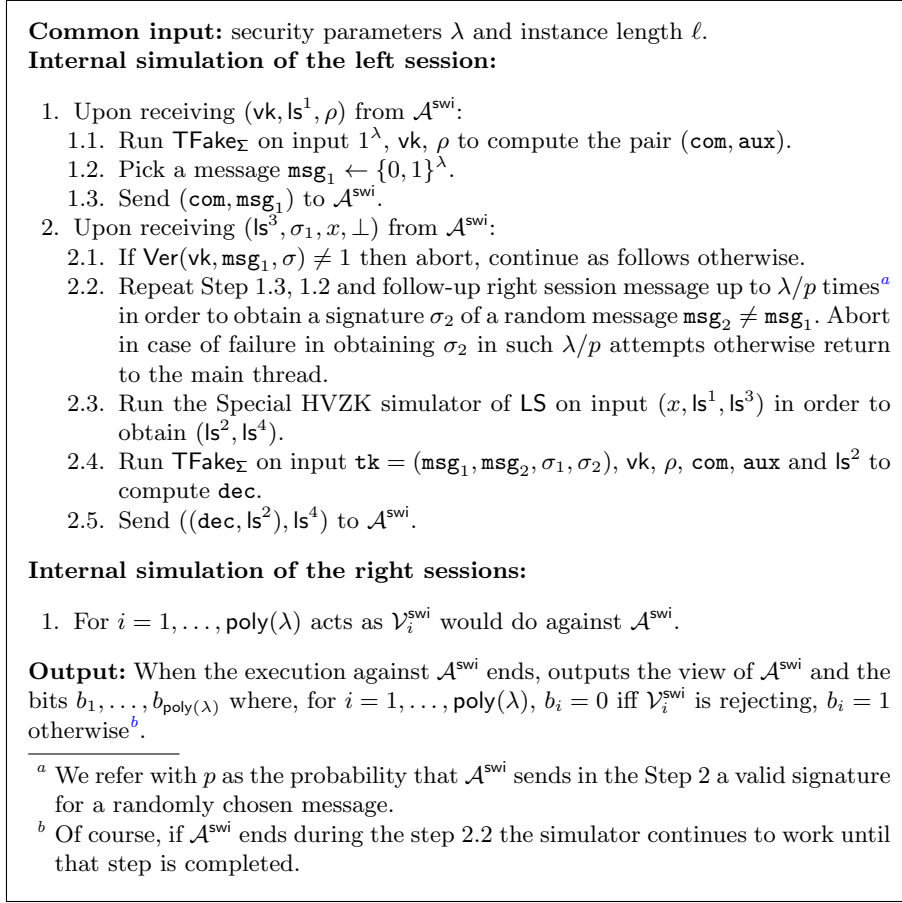
In the security proof we denote by  $\{\text{wimim}_{\mathcal{H}_i}(1^\lambda, z)\}_{\lambda \in \mathbb{N}, z \in \{0,1\}^*}$  the random variable describing 1) the view of  $\mathcal{A}^{\text{swi}}$ , 2) the output of  $\mathcal{V}_i^{\text{swi}}$  for  $i = 1, \dots, \text{poly}(\lambda)$ , 3) the first  $\gamma$ -bits  $\tilde{w}_1^\gamma, \dots, \tilde{w}_{\text{poly}(\lambda)}^\gamma$  of the corresponding witnesses  $\tilde{w}_1, \dots, \tilde{w}_{\text{poly}(\lambda)}$  w.r.t. the instances  $\tilde{x}_1, \dots, \tilde{x}_{\text{poly}(\lambda)}$  that appear in  $\mathcal{A}^{\text{swi}}$ 's view except that  $\tilde{w}_i^\gamma = \perp$  if  $\mathcal{V}_i^{\text{swi}}$  rejected, with  $i = 1, \dots, \text{poly}(\lambda)$  <sup>21</sup>.

The proof makes use of the following main hybrid experiments.

- The 1st hybrid experiment is  $\mathcal{H}_1(1^\lambda, z)$ . In this hybrid in the left session  $\mathcal{P}^{\text{swi}}$  interacts with  $\mathcal{A}^{\text{swi}}$  in order to prove the validity of the instance  $x$  using the witness  $w$ , while in the right sessions  $\mathcal{V}_i^{\text{swi}}$  interacts with  $\mathcal{A}^{\text{swi}}$  for  $i = 1, \dots, \text{poly}(\lambda)$ . We want to prove that in the  $i$ -th right session  $\mathcal{A}^{\text{swi}}$  does not prove any false instance  $\tilde{x}_i$  for any  $i = 1, \dots, \text{poly}(\lambda)$  <sup>22</sup>. This property follows immediately from the adaptive-input AoK of SWI. We observe that in this case it is crucial that SWI is adaptive-input AoK, because we are considering an adversary  $\mathcal{A}^{\text{swi}}$  that can choose the instance to be proved in the last round of every right session.
- The 2nd hybrid experiment is  $\mathcal{H}_2(1^\lambda, z)$  and differs from  $\mathcal{H}_1(1^\lambda, z)$  in the way the commitment  $\text{com}$  and the decommitment information  $\text{dec}$  are computed in the left session. More precisely,  $\mathcal{P}^{\text{swi}}$  runs  $\text{TFake}_\Sigma$  to compute a commitment  $(\rho, \text{com})$ , and subsequently to compute a decommitment of  $(\rho, \text{com})$  to the value  $\text{ls}^2$  (we remark that no trapdoor is needed to run  $\text{TFake}_\Sigma$  in order to compute  $(\rho, \text{com})$ ). In more details, this experiment rewinds the adversary  $\mathcal{A}^{\text{swi}}$  from the 3rd to the 2nd round of the left session to extract two signatures  $\sigma_1, \sigma_2$  of two different messages  $(\text{msg}_1, \text{msg}_2)$  and uses them as trapdoor to run  $\text{TFake}_\Sigma$ . The indistinguishability between  $\text{wimim}_{\mathcal{H}_1}(1^\lambda, z)$  and  $\text{wimim}_{\mathcal{H}_2}(1^\lambda, z)$  comes from the hiding and the trapdooriness of  $\text{TC}_\Sigma$ .
- The 3rd hybrid experiment is  $\mathcal{H}_3(1^\lambda, z)$  and differs from  $\mathcal{H}_2(1^\lambda, z)$  in the way the transcript for LS is computed. In more details the Special HVZK

<sup>21</sup> To ease the notation sometimes we will refer to  $\{\text{wimim}_{\mathcal{H}_i}(1^\lambda, z)\}_{\lambda \in \mathbb{N}, z \in \{0,1\}^*}$  using just  $\text{wimim}_{\mathcal{H}_i}(1^\lambda, z)$ .

<sup>22</sup> When we refer to a *proved instance*  $\tilde{x}_i$  we implicitly assume that  $\mathcal{V}_i^{\text{swi}}$  is accepting, with  $i = 1, \dots, \text{poly}(\lambda)$ .

Fig. 3: The SimWI  $\mathcal{S}^{\text{swi}}$  for SWI.

simulator  $\mathcal{S}$  of LS is used to compute the messages  $ls^2$  and  $ls^4$  instead of using the honest procedure  $\mathcal{P}^{\text{swi}}$ . The indistinguishability between  $\text{wimim}_{\mathcal{H}_2}(1^\lambda, z)$  and  $\text{wimim}_{\mathcal{H}_3}(1^\lambda, z)$  comes from the Special HVZK of LS. We observe that the security proof ends with this hybrid experiment because  $\text{wimim}_{\mathcal{H}_3}(1^\lambda, z) \equiv \text{sim}_{\text{SWI}}^{\text{swi}}(1^\lambda, z)$ .

A formal proof is given in the full version (see [5]).

## 4 4-Round Concurrent NM Commitment Scheme

Our construction makes use of an adaptive-input SimWI AoK  $\text{SWI} = (\mathcal{P}^{\text{swi}}, \mathcal{V}^{\text{swi}})$  combined with a weak concurrent NM commitment scheme  $\Pi_{\text{wom}}$ . For our propose we consider a weak NM commitment scheme that with overwhelming probability any well-formed commitment can be opened to only one message. We recall

that the weak concurrent NM commitment scheme of [25] enjoys this property when instantiated with Naor’s commitment scheme [35].

We now consider the following language  $L$  based on the weak NM commitment scheme  $\Pi_{\text{wom}} = (\text{Sen}_{\text{wom}}, \text{Rec}_{\text{wom}})$ :

$$L = \{(\tau, \text{id}) : \exists (m, \text{dec}) \text{ s.t. } \text{Rec}_{\text{wom}} \text{ on input } (m, \text{dec}, \text{id}) \text{ accepts } m \text{ as decommitment of } \tau\}$$

and the corresponding relation  $\text{Rel}_L$ .

We now use  $\text{SWI} = (\mathcal{P}^{\text{swi}}, \mathcal{V}^{\text{swi}})$  to upgrade a 4-round public-coin concurrent weak NM commitment scheme  $\Pi_{\text{wom}}$  with the property that after the second round there is at most one valid message, to a concurrent NM commitment scheme. We will be able to invoke the security of SWI, since the language  $L$  is a  $\gamma$ -prefix language with overwhelming probability with  $\gamma = |m|$ . In fact, given an instance  $(\tau, \text{id})$  of  $L$  all the witnesses  $w_1, \dots, w_n$  of  $(\tau, \text{id})$  have the form  $m|\text{dec}_i$  for  $i = 1, \dots, n$  (i.e., all witnesses have the same prefix  $m$ ). Consider a SimWI AoK for  $L$ . Let  $m$  be the message that NM4Sen wants to commit and  $\text{id}$  be the id for this session. The high-level idea of our protocol is depicted in Fig. 4.

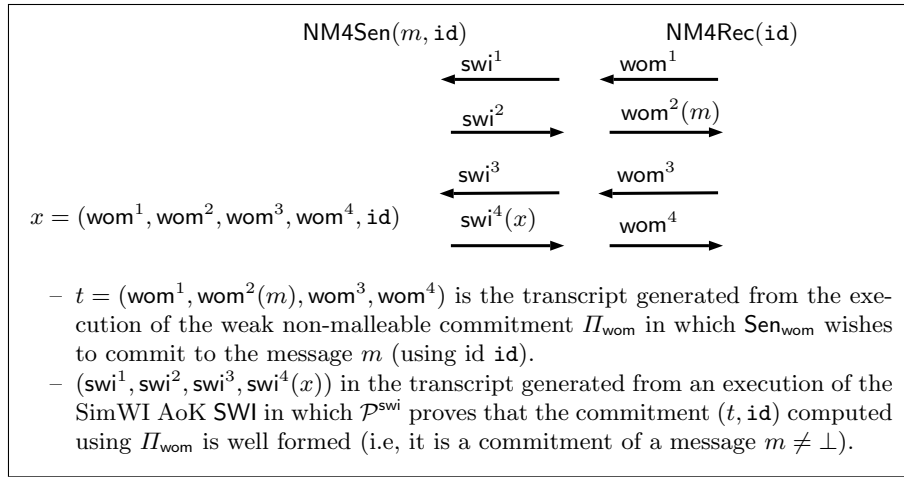


Fig. 4: 4-Round Concurrent NM Commitment Scheme from OWFs.

In the 1st round the receiver NM4Rec computes and sends the 1st round  $\text{swi}^1$  of SWI and the 1st round  $\text{wom}^1$  of  $\Pi_{\text{wom}}$  using as input the  $\text{id}$ . Then NM4Sen on input  $\text{id}$ , the message  $m$  and the received 1st round, computes the 2nd round  $\text{wom}^2$  of  $\Pi_{\text{wom}}$  in order to commit to the message  $m$ , using  $\text{id}$ , furthermore he obtains  $\text{dec}_{\text{wom}}$  s.t.  $(m, \text{dec}_{\text{wom}})$  constitutes the decommitment information<sup>23</sup>.

<sup>23</sup> In order to match the adaptive-input selection satisfied by SWI, message and randomness explaining the entire transcript are already fixed in this round.



Moreover NM4Sen computes and sends the 2nd round  $\text{swi}^2$  of SWI. In the 3rd round NM4Rec sends the 3rd round  $\text{wom}^3$  of  $\Pi_{\text{wom}}$  and the 3rd round  $\text{swi}^3$  of SWI. In the last round NM4Sen computes the 4th round  $\text{wom}^4$  of  $\Pi_{\text{wom}}$ . Furthermore, NM4Sen, using  $(\text{wom}^1, \text{wom}^2, \text{wom}^3, \text{wom}^4, \text{id})$  as instance and  $(m, \text{dec}_{\text{wom}})$  as a witness, computes the 4th round  $\text{swi}^4$  of SWI and sends  $\text{wom}^4, \text{swi}^4$  to NM4Rec. At this point NM4Rec accepts the commitment (i.e., the transcript of the protocol generated so far) iff the transcript for SWI is accepting for  $\mathcal{V}^{\text{swi}}$  with respect to the instance  $(\text{wom}^1, \text{wom}^2, \text{wom}^3, \text{wom}^4, \text{id})$ . The decommitment phase of our scheme simply corresponds to the decommitment phase of  $\Pi_{\text{wom}}$ .

As described before,  $\text{SWI} = (\mathcal{P}^{\text{swi}}, \mathcal{V}^{\text{swi}})$  is used by NM4Sen to prove knowledge of a message and randomness consistent with the transcript computed using  $\Pi_{\text{wom}}$ . To execute SWI the instance is not needed until the last round.

Fig. 5 describes in details our 4-round concurrent NM commitment scheme  $\Pi_{\text{NM4Com}}$ .

**Theorem 2.** *Assuming OWFs,  $\Pi_{\text{NM4Com}} = (\text{NM4Sen}, \text{NM4Rec})$  is a 4-round concurrent NM commitment scheme.*

The 4-round concurrent NM commitment scheme  $\Pi_{\text{NM4Com}} = (\text{NM4Sen}, \text{NM4Rec})$  relies on OWFs, because the adaptive-input SimWI AoK SWI can be constructed using OWFs only (see Theorem 1). Furthermore  $\Pi_{\text{wom}}$  can be instantiated using the weak one-one non-malleable commitment scheme of [25] that is proved to be weak concurrent non-malleable in the full version of our work (see [5]). Note that this construction relies on OWFs and has also the additional property that we require (i.e. after the second round the only valid message and the corresponding decommitment informations are fixed). The security proof is divided in two parts. In the 1st part we prove that  $\Pi_{\text{NM4Com}}$  is indeed a commitment scheme. In the second part we prove that  $\Pi_{\text{NM4Com}}$  is a one-many NM commitment scheme, and then we go from one-many to concurrent non-malleability by using Proposition 1.

**Lemma 2.**  *$\Pi_{\text{NM4Com}} = (\text{NM4Sen}, \text{NM4Rec})$  is a statistically binding computationally hiding commitment scheme.*

*Proof. Correctness.* The correctness follows directly from the delayed-input completeness of SWI and the correctness of  $\Pi_{\text{wom}}$ .

**Statistically Binding.** Observe that the message given in output in the decommitment phase of  $\Pi_{\text{NM4Com}}$  is the message committed using  $\Pi_{\text{wom}}$ . Moreover the decommitment of  $\Pi_{\text{NM4Com}}$  coincides with the decommitment of  $\Pi_{\text{wom}}$ . Since  $\Pi_{\text{wom}}$  is statistically binding then so is  $\Pi_{\text{NM4Com}}$ .

**Computationally Hiding.** Computational hiding follows immediately from Lemma 3.

**Lemma 3.** *For all  $m \in \{0, 1\}^{\text{poly}(\lambda)}$   $\{\text{mim}_{\Pi_{\text{NM4Com}}}^{\mathcal{A}_{\text{NM4Com}}, m}(z)}\}_{z \in \{0, 1\}^*} \approx \{\text{sim}_{\Pi_{\text{NM4Com}}}^{\mathcal{S}_{\text{NM4Com}}}(1^\lambda, z)\}_{z \in \{0, 1\}^*}$ .*

We denote by  $\{\text{mim}_{\mathcal{H}_i^m}^{\mathcal{A}_{\text{NM4Com}}, m}(z)\}_{z \in \{0, 1\}^*}$  the random variable describing the view of the MiM  $\mathcal{A}_{\text{NM4Com}}$  combined with the values that it commits in the the  $\text{poly}(\lambda)$  right sessions in hybrid  $\mathcal{H}_i^m(z)$ .

**Common input:** security parameter  $\lambda$ , instance length  $\ell$ , NM4Sen's identity  $\text{id} \in \{0, 1\}^\lambda$ .

**Input to NM4Sen:**  $m \in \{0, 1\}^{\text{poly}\{\lambda\}}$ .

**Commitment phase:**

1. NM4Rec  $\rightarrow$  NM4Sen
  - 1.1. Run  $\text{Rec}_{\text{wom}}$  on input  $1^\lambda, \text{id}$  thus obtaining the 1st round  $\text{wom}^1$  of  $\Pi_{\text{wom}}$ .
  - 1.2. Run  $\mathcal{V}^{\text{swi}}$  on input  $1^\lambda$  and  $\ell$  thus obtaining the 1st round  $\text{swi}^1$  of SWI.
  - 1.3. Send  $(\text{swi}^1, \text{wom}^1)$  to NM4Sen.
2. NM4Sen  $\rightarrow$  NM4Rec
  - 2.1. Run  $\mathcal{P}^{\text{swi}}$  on input  $1^\lambda, \ell$  and  $\text{swi}^1$  thus obtaining the 2nd round  $\text{swi}^2$  of SWI.
  - 2.2. Run  $\text{Sen}_{\text{wom}}$  on input  $1^\lambda, \text{id}, \text{wom}^1$  and the message  $m$  thus obtaining the 2nd round  $\text{wom}^2$  of  $\Pi_{\text{wom}}$  and  $\text{dec}_{\text{wom}}$  s.t.  $(m, \text{dec}_{\text{wom}})$  constitutes the decommitment information.
  - 2.3. Send  $(\text{swi}^2, \text{wom}^2)$  to NM4Rec.
3. NM4Rec  $\rightarrow$  NM4Sen
  - 3.1. Run  $\text{Rec}_{\text{wom}}$  on input  $\text{wom}^2$  thus obtaining the 3rd round  $\text{wom}^3$  of  $\Pi_{\text{wom}}$ .
  - 3.2. Run  $\mathcal{V}^{\text{swi}}$  on input  $\text{swi}^2$  thus obtaining the 3rd round  $\text{swi}^3$  of SWI.
  - 3.3. Send  $(\text{wom}^3, \text{swi}^3)$  to NM4Sen.
4. NM4Sen  $\rightarrow$  NM4Rec
  - 4.1. Run  $\text{Sen}_{\text{wom}}$  on input  $\text{wom}^3$  thus obtaining the 4th round  $\text{wom}^4$  of  $\Pi_{\text{wom}}$ .
  - 4.2. Set  $x = (\text{wom}^1, \text{wom}^2, \text{wom}^3, \text{wom}^4, \text{id})$  and  $w = (m, \text{dec}_{\text{wom}})$  with  $|x| = \ell$ . Run  $\mathcal{P}^{\text{swi}}$  on input  $x, w$  and  $\text{swi}^3$  thus obtaining the 4th round  $\text{swi}^4$  of SWI.
  - 4.3. Send  $(\text{wom}^4, \text{swi}^4)$  to NM4Rec.
5. NM4Rec: Set  $x = (\text{wom}^1, \text{wom}^2, \text{wom}^3, \text{wom}^4, \text{id})$  and accept the commitment iff  $(\text{swi}^1, \text{swi}^2, \text{swi}^3, \text{swi}^4)$  is accepting for  $\mathcal{V}^{\text{swi}}$  with respect to the instance  $x$ .

**Decommitment phase:**

1. NM4Sen  $\rightarrow$  NM4Rec: Send  $(m, \text{dec}_{\text{wom}})$  to NM4Rec.
2. NM4Rec: accept  $m$  as the committed message if and only if  $\text{Rec}_{\text{wom}}$ , on input  $(m, \text{dec}_{\text{wom}})$ , accepts  $m$  as the committed message of  $(\text{wom}^1, \text{wom}^2, \text{wom}^3, \text{wom}^4, \text{id})$ .

Fig. 5: 4-round Concurrent NM Commitments  $\Pi_{\text{NM4Com}}$  from OWFs.

As required by the definition, we want to show that the distribution of the real game experiment (i.e., the view of the MiM  $\mathcal{A}^{\text{NM4Com}}$  when playing with NM4Sen committing  $m$  along with the messages committed in the right sessions) and the one of the output of a simulator are computationally indistinguishable. We start by showing the simulator  $\mathcal{S}^{\text{NM4Com}}$  and giving an overview of the entire proof. The simulator is described in Figure 6.

- The 1st hybrid experiment is  $\mathcal{H}_1^m(z)$ . In this hybrid in the left session NM4Sen commits to  $m$ , while in the right sessions  $\text{NM4Rec}_i$  interacts with  $\mathcal{A}^{\text{NM4Com}}$  for  $i = 1, \dots, \text{poly}(\lambda)$ . We prove that in the  $i$ -th right session

**Common input:** Security parameters:  $\lambda$ . NM4Sen's identity:  $\text{id} \in \{0, 1\}^\lambda$ .

**Internal simulation of the left session:**

1. Upon receiving  $(\text{swi}^1, \text{wom}^1)$  from  $\mathcal{A}^{\text{NM4Com}}$ :
  - 1.1. Run  $\mathcal{P}^{\text{swi}}$  on input  $1^\lambda, \ell$  and  $\text{swi}^1$  thus obtaining the 2nd round  $\text{swi}^2$  of SWI.
  - 1.2. Run  $\text{Sen}_{\text{wom}}$  on input  $1^\lambda, \text{id}, \text{wom}^1$  and the message  $0^\lambda$  thus obtaining the 2nd round  $\text{wom}^2$  of  $\Pi_{\text{wom}}$  and  $\text{dec}_{\text{wom}}$  s.t.  $(0^\lambda, \text{dec}_{\text{wom}})$  constitutes the decommitment informations.
  - 1.3. Send  $(\text{swi}^2, \text{wom}^2)$  to  $\mathcal{A}^{\text{NM4Com}}$ .
2. Upon receiving  $(\text{wom}^3, \text{swi}^3)$  from  $\mathcal{A}^{\text{NM4Com}}$ :
  - 2.1. Run  $\text{Sen}_{\text{wom}}$  on input  $\text{wom}^3$  thus obtaining the 4th round  $\text{wom}^4$  of  $\Pi_{\text{wom}}$ .
  - 2.2. Set  $x = (\text{wom}^1, \text{wom}^2, \text{wom}^3, \text{wom}^4, \text{id})$  and  $w = (0^\lambda, \text{dec}_{\text{wom}})$  with  $|x| = \ell$ . Run  $\mathcal{P}^{\text{swi}}$  on input  $x, w$  and  $\text{swi}^3$  thus obtaining the 4th round  $\text{swi}^4$  of SWI.
  - 2.3. Send  $(\text{wom}^4, \text{swi}^4)$  to  $\mathcal{A}^{\text{NM4Com}}$ .

**Right sessions:**

1.  $S^{\text{NM4Com}}$  acts as a proxy between  $\mathcal{A}^{\text{NM4Com}}$  and  $\text{NM4Rec}_i$ , with  $i = 1, \dots, \text{poly}(\lambda)$ .

Fig. 6: The simulator  $S^{\text{NM4Com}}$  of  $\Pi_{\text{NM4Com}}$ .

$\mathcal{A}^{\text{NM4Com}}$  does not commit to a message  $\tilde{m}_i = \perp$  for any  $i = 1, \dots, \text{poly}(\lambda)$ . The proof follows immediately from the adaptive-input AoK of SWI. We observe that in this case it is crucial that SWI is adaptive-input AoK, because the theorem proved by  $\mathcal{A}^{\text{NM4Com}}$  are fully specified only in the last round of every right session. Clearly we have that  $\text{mim}_{\Pi_{\text{NM4Com}}}^{\mathcal{A}^{\text{NM4Com}}, m}(z) = \text{mim}_{\mathcal{H}_1^m}^{\mathcal{A}^{\text{NM4Com}}, m}(z)$ .

- The 2nd hybrid experiment is  $\mathcal{H}_2^m(z)$  and differs from  $\mathcal{H}_1^m(z)$  in the way the transcript of SWI is computed. In this hybrid the simulator  $\mathcal{S}^{\text{swi}}$  of SWI is used to compute the transcript of SWI. The indistinguishability between  $\text{mim}_{\mathcal{H}_1^m}^{\mathcal{A}^{\text{NM4Com}}, m}(z)$  and  $\text{mim}_{\mathcal{H}_2^m}^{\mathcal{A}^{\text{NM4Com}}, m}(z)$  comes from the adaptive-input SimWI property of SWI. It is important to observe that we can properly rely on the adaptive-input SimWI property of SWI since the committed message in  $\Pi_{\text{wom}}$  is fixed in the second round. Therefore also the family of statement  $\mathcal{X}_w = \{x : (x, w) \in \text{Rel}_L \text{ or } x \notin L\}$  proved using SWI is implicitly fixed in the second round. Moreover we can rely on the security of SWI because the language  $L$  is a  $\gamma$ -prefix language for  $\text{prefix} = |m|$ . Indeed, all witnesses of any instance of  $L$  have the same prefix (i.e., the committed message  $m$ ). Therefore when using the simulator of SWI we are guaranteed that the distribution of the first  $\gamma$  bits of the witnesses corresponding to the statements proven by the adversary in the right sessions of SWI does not change. In turn, this implies that the distribution of the committed messages in the right sessions does not change since each message committed in a session is in the first  $\gamma$

bits of any witness corresponding to the statement proven in SWI in that session.

We also consider the hybrid experiments  $\mathcal{H}_1^0(z)$ ,  $\mathcal{H}_2^0(z)$ , that are the same hybrid experiments described above with the difference that  $\Pi_{\text{wom}}$  is used to commit to a message  $0^\lambda$  instead of  $m$ . From the same arguments described above we have that  $\text{mim}_{\mathcal{H}_1^0}^{\mathcal{A}^{\text{NM4Com}}, m}(z) \approx \text{mim}_{\mathcal{H}_2^0}^{\mathcal{A}^{\text{NM4Com}}, m}(z)$  and that in the  $i$ -th right session of  $\mathcal{H}_1^0(z)$   $\mathcal{A}^{\text{NM4Com}}$  commits to a message  $\tilde{m}_i = \perp$  with negligible probability (for any  $i = 1, \dots, \text{poly}(\lambda)$ ). We also observe that  $\text{mim}_{\mathcal{H}_1^0}^{\mathcal{A}^{\text{NM4Com}}, m}(z) = \text{sim}_{\Pi_{\text{NM4Com}}}^{\mathcal{S}^{\text{NM4Com}}}(1^\lambda, z)$ .

The only thing that remains to argue to complete the proof is that the view of  $\mathcal{A}^{\text{NM4Com}}$ , along with messages committed in the right sessions of the execution of  $\mathcal{H}_2^m(z)$ , is indistinguishable from the view of  $\mathcal{A}^{\text{NM4Com}}$  along with the messages committed in the right sessions of  $\mathcal{H}_2^0(z)$ . This is actually ensured by the weak concurrent non-malleability of  $\Pi_{\text{wom}}$ . Indeed, from the arguments given above, in both  $\mathcal{H}_2^m(z)$  and  $\mathcal{H}_2^0(z)$  the adversary  $\mathcal{A}^{\text{NM4Com}}$  commits to a message  $\tilde{m}_i = \perp$  with negligible probability for  $i = 1, \dots, \text{poly}(\lambda)$ . Therefore we can use this  $\mathcal{A}^{\text{NM4Com}}$  to construct an adversary  $\mathcal{A}^{\text{wom}}$  that breaks the weak concurrent non-malleability of  $\Pi_{\text{wom}}$ . Roughly speaking, let  $m, 0^\lambda$  be the challenge messages, then  $\mathcal{A}^{\text{wom}}$  works as following against the challenger  $\mathcal{C}^{\text{wom}}$ . In the left session acts as a proxy for all the messages of  $\Pi_{\text{wom}}$  between  $\mathcal{C}^{\text{wom}}$  and  $\mathcal{A}^{\text{NM4Com}}$  and executes the simulator  $\mathcal{S}^{\text{swi}}$  of SWI in parallel. In the  $i$ -th right session  $\mathcal{A}^{\text{wom}}$  interacts as  $\text{Rec}_{\text{wom}, i}$  would do w.r.t. the messages of  $\Pi_{\text{wom}}$  and as  $\mathcal{V}_i^{\text{swi}}$  for the messages of SWI, for all  $i = 1, \dots, \text{poly}(\lambda)$ . The distinguisher that break the concurrent weak non-malleability of  $\Pi_{\text{wom}}$  runs  $\mathcal{D}^{\text{NM4Com}}$  (that exists by contradiction) that distinguishes  $\text{mim}_{\mathcal{H}_2^0}^{\mathcal{A}^{\text{NM4Com}}, m}(z)$  from  $\text{mim}_{\mathcal{H}_2^m}^{\mathcal{A}^{\text{NM4Com}}, m}(z)$ , and outputs what  $\mathcal{D}^{\text{NM4Com}}$  outputs.

A caveat that we have to address in this reduction is due to the rewinds made by  $\mathcal{S}^{\text{swi}}$  in the left session in order to compute the transcript of SWI. Indeed a rewind made in the left session could affect the reduction rewinding also the receivers of  $\Pi_{\text{wom}}$  involved in the reduction. More precisely could happen that in a session  $j \in \{1, \dots, \text{poly}(\lambda)\}$  the third round of  $\Pi_{\text{wom}}$  has to be played multiple times because of the multiple values  $\tilde{\text{wom}}_j^2$  received in the  $j$ -th right session. We can avoid this problem by sending a random string as a third round of  $\Pi_{\text{wom}}$ . In this way for the first value  $\tilde{\text{wom}}_j^2$  received from  $\mathcal{A}^{\text{NM4Com}}$  the reduction interacts with the receiver of  $\Pi_{\text{wom}}$  and for all the other values the reduction sends a random string. This is the reason why in our construction we require  $\Pi_{\text{wom}}$  to be public coin. One additional issue is the following. The simulator of SWI could rewind the entire left session during the reduction, therefore requiring to compute a new commitment of  $m$  for the protocol  $\Pi_{\text{wom}}$ . Since we are assuming that  $\Pi_{\text{wom}}$  is weak concurrent non-malleable, the reduction can request to receive multiple commitments for the same message. A formal proof is given in the full version (see [5]).

## 5 3-Round NM Commitments from Strong OWPs

In addition to the definition of NM commitments given in Section 2.3, in this section we consider also a synchronous NM commitment scheme secure against a sub-exponential time adversary. The definitions follow below.

### 5.1 Synchronous NM Commitment Scheme

**Definition 7** (*synchronous NM commitment scheme*). *A commitment scheme is synchronous one-one (resp., one-many) non-malleable if it is one-one (resp., one-many) NM with respect to synchronous MiM adversaries.*

We also consider the definition of a NM commitment scheme secure against a MIM  $\mathcal{A}$  running in time bounded by  $T = 2^{\lambda^\alpha}$  for some positive constant  $\alpha < 1$ . In this case we will say that a commitment scheme is  $T$ -non-malleable. We will also say that an NM commitment scheme is  $\tilde{T}$ -breakable to specify that an algorithm which runs in time  $\tilde{T} = 2^{\lambda^\beta}$ , for some positive constant  $\beta < 1$ , can maul the committed message.

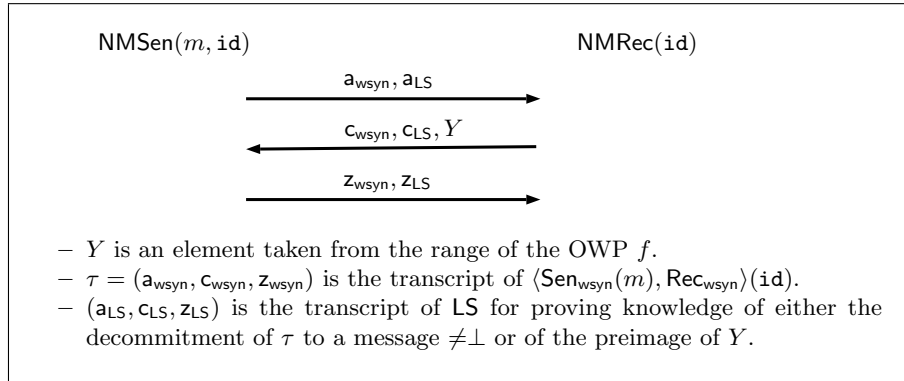


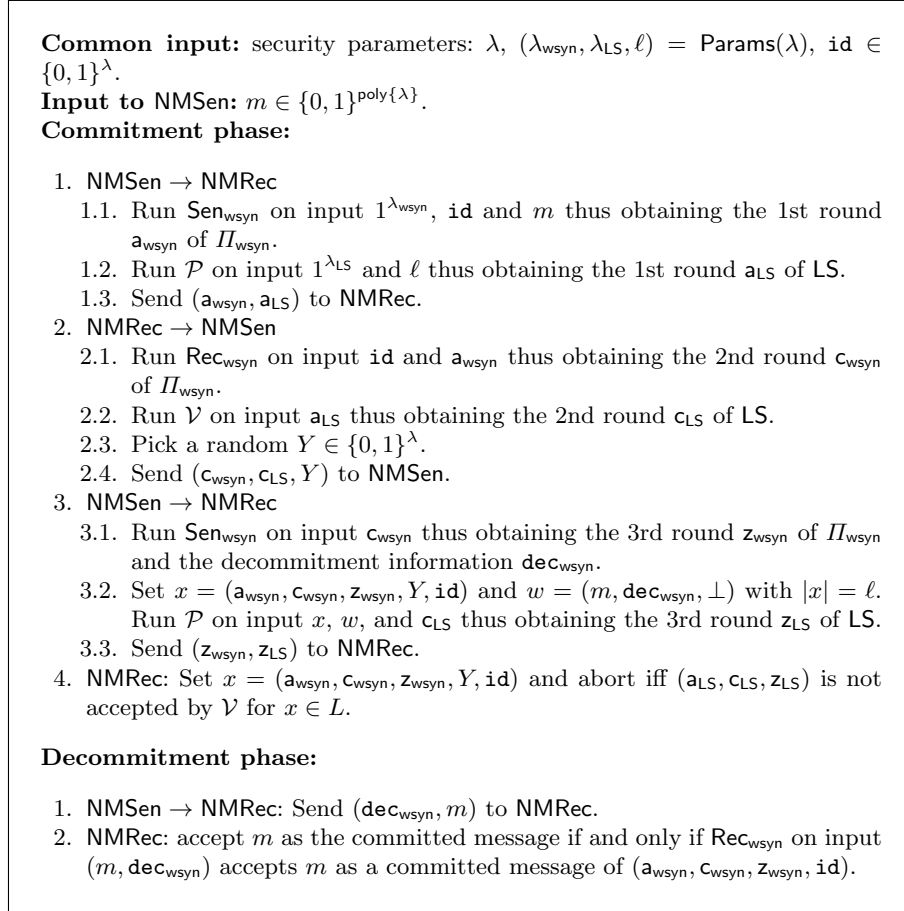
Fig. 7: Informal description of our 3-round NM commitment scheme  $\Pi_{\text{NMCom}}$ .

### 5.2 3-Round NM Commitment Scheme: $\Pi_{\text{NMCom}} = (\text{NMSen}, \text{NMRec})$

Our construction is based on a compiler that takes as input a 3-round synchronous weak one-one NM commitment scheme  $\Pi_{\text{wsyn}} = (\text{Sen}_{\text{wsyn}}, \text{Rec}_{\text{wsyn}})$ , a OWP  $f$ , a WI adaptive PoK for  $\mathcal{NP}$  LS, and outputs a 3-round extractable one-one NM commitment scheme  $\Pi_{\text{NMCom}} = (\text{NMSen}, \text{NMRec})$ .

In order to construct our compiler we consider the following tools:

1. a OWP  $f$  that is secure against PPT adversaries and that is  $\tilde{T}_f$ -breakable;

Fig. 8: 3-Round NM Commitment scheme  $\Pi_{\text{NMCom}}$ .

2. a 3-round one-one synchronous weak NM commitment scheme  $\Pi_{\text{wsyn}} = (\text{Sen}_{\text{wsyn}}, \text{Rec}_{\text{wsyn}})$  that is  $T_{\text{wsyn}}$ -hiding/NM, and  $\tilde{T}_{\text{wsyn}}$ -breakable;
3. the LS PoK  $\text{LS} = (\mathcal{P}, \mathcal{V})$  for the language

$$L = \{(a, c, z, Y, \text{id}) : \exists (m, \text{dec}, y) \text{ s.t. } (\text{Rec}_{\text{wsyn}} \text{ on input } (a, c, z, m, \text{dec}, \text{id}) \text{ accepts } m \neq \perp \text{ as a decommitment of } (a, c, z, \text{id}) \text{ OR } Y = f(y))\}$$

that is  $T_{\text{LS}}$ -WI for the corresponding relation  $\text{Rel}_L$ .

Let  $\lambda$  be the security parameter of our scheme. We use w.l.o.g.  $\lambda$  also as security parameter for the one-wayness of  $f$  with respect to polynomial-time adversaries. We consider the following hierarchy of security levels:  $\tilde{T}_f \ll T_{\text{wsyn}} \ll \tilde{T}_{\text{wsyn}} = \sqrt{T_{\text{LS}}} \ll T_{\text{LS}}$  where by “ $T \ll T'$ ” we mean that “ $T \cdot \text{poly}(\lambda) < T'$ ”.

Now, similarly to [47, 6], we define different security parameters, one for each tool involved in the security proof to be consistent with the hierarchy of security

levels defined above. Given the security parameter  $\lambda$  of our scheme, we will make use of the following security parameters: 1)  $\lambda$  for the OWP  $f$ ; 2)  $\lambda_{\text{wsyn}}$  for the synchronous weak one-one NM commitment scheme; 3)  $\lambda_{\text{LS}}$  for LS. A formal proof of the following theorem is given in the full version (see [5]).

**Theorem 3.** *Suppose there exists a synchronous weak one-one NM commitment scheme and OWPs, both secure against subexponential-time adversaries, then  $\Pi_{\text{NMCom}}$  is an extractable one-one NM commitment scheme.*

## 6 Acknowledgments

We thank Vipul Goyal, and Silas Richelson for remarkable discussions on [24]. We thank Giuseppe Persiano and Alessandra Scafuro for several discussions on delayed-input protocols. We also thank to an anonymous reviewer of CRYPTO 2017 for suggesting prefix languages for SimWI. Research supported in part by “GNCS - INdAM”, EU COST Action IC1306, NSF grants 1065276, 1118126 and 1136174, US-Israel BSF grant 2008411, OKAWA Foundation Research Award, IBM Faculty Research Award, Xerox Faculty Research Award, B. John Garrick Foundation Award, Teradata Research Award, and Lockheed-Martin Corporation Research Award. This material is based upon work supported in part by DARPA Safeware program. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. The work of 1st, 3rd and 4th authors has been done in part while visiting UCLA.

## References

1. BARAK, B. Constant-round coin-tossing with a man in the middle or realizing the shared random string model. In *43rd Symposium on Foundations of Computer Science (FOCS 2002)*, pp. 345–355.
2. CAO, Z., VISCONTI, I., AND ZHANG, Z. Constant-round concurrent non-malleable statistically binding commitments and decommitments. In *Public Key Cryptography - PKC 2010, 13th International Conference on Practice and Theory in Public Key Cryptography. Proceedings* (2010), vol. 6056 of *LNCS*, Springer, pp. 193–208.
3. CHUNG, K., OSTROVSKY, R., PASS, R., VENKITASUBRAMANIAM, M., AND VISCONTI, I. 4-round resettable-sound zero knowledge. In *TCC 2014. Proceedings* (2014), vol. 8349 of *LNCS*, Springer, pp. 192–216.
4. CHUNG, K., PASS, R., AND SETH, K. Non-black-box simulation from one-way functions and applications to resettable security. In *Symposium on Theory of Computing Conference, STOC 2013*, ACM, pp. 231–240.
5. CIAMPI, M., OSTROVSKY, R., SINISCALCHI, L., AND VISCONTI, I. 4-round concurrent non-malleable commitments from one-way functions. Cryptology ePrint Archive, Report 2016/621, 2016. <http://eprint.iacr.org/2016/621>.
6. CIAMPI, M., OSTROVSKY, R., SINISCALCHI, L., AND VISCONTI, I. Concurrent non-malleable commitments (and more) in 3 rounds. In *CRYPTO 2016, Proceedings, Part III* (2016), vol. 9816 of *LNCS*, Springer, pp. 270–299.



7. CIAMPI, M., PERSIANO, G., SCAFURO, A., SINISCALCHI, L., AND VISCONTI, I. Improved or-composition of sigma-protocols. In *TCC 2016-A, Proceedings, Part II* (2016), vol. 9563 of *LNCS*, Springer, pp. 112–141.
8. CIAMPI, M., PERSIANO, G., SCAFURO, A., SINISCALCHI, L., AND VISCONTI, I. Online/offline OR composition of sigma protocols. In *Advances in Cryptology - EUROCRYPT 2016 Proceedings, Part II* (2016), vol. 9666 of *LNCS*, Springer, pp. 63–92.
9. DACHMAN-SOLED, D., MALKIN, T., RAYKOVA, M., AND VENKITASUBRAMANIAM, M. Adaptive and concurrent secure computation from new adaptive, non-malleable commitments. In *A ASIACRYPT 2013 Proceedings, Part I* (2013), pp. 316–336.
10. DAMGÅRD, I., AND GROTH, J. Non-interactive and reusable non-malleable commitment schemes. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, June 9-11, 2003, San Diego, CA, USA* (2003), pp. 426–437.
11. DI CRESCENZO, G., PERSIANO, G., AND VISCONTI, I. Constant-round resettable zero knowledge with concurrent soundness in the bare public-key model. In *CRYPTO 2004, Proceedings*, (2004), vol. 3152 of *LNCS*, Springer, pp. 237–253.
12. DOLEV, D., DWORK, C., AND NAOR, M. Non-malleable cryptography (extended abstract). In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA* (1991), pp. 542–552.
13. FEIGE, U. Alternative models for zero knowledge interactive proofs. Master’s thesis, Weizmann Institute of Science, Rehovot, Israel, 1990. Ph.D. thesis.
14. FEIGE, U., LAPIDOT, D., AND SHAMIR, A. Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In *31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Vol. I* (1990), IEEE Computer Society, pp. 308–317.
15. FEIGE, U., AND SHAMIR, A. Witness indistinguishable and witness hiding protocols. In *Proceedings of the Twenty-second Annual ACM Symposium on Theory of Computing* (New York, NY, USA, 1990), STOC ’90, ACM, pp. 416–426.
16. GARG, S., MUKHERJEE, P., PANDEY, O., AND POLYCHRONIADOU, A. The exact round complexity of secure computation. In *Advances in Cryptology - EUROCRYPT 2016, Proceedings, Part II* (2016), vol. 9666 of *LNCS*, Springer, pp. 448–476.
17. GOLDREICH, O., AND KRAWCZYK, H. On the composition of zero-knowledge proof systems. *SIAM J. Comput.* 25, 1 (1996), 169–192.
18. GOLDREICH, O., AND LEVIN, L. A. A hard-core predicate for all one-way functions. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washington, USA* (1989), pp. 25–32.
19. GOYAL, V. Constant round non-malleable protocols using one way functions. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011* (2011), pp. 695–704.
20. GOYAL, V., JAIN, A., OSTROVSKY, R., RICHELSON, S., AND VISCONTI, I. Constant-round concurrent zero knowledge in the bounded player model. In *ASIACRYPT 2013, Proceedings* (2013), vol. 8279 of *LNCS*, Springer, pp. 21–40.
21. GOYAL, V., KHURANA, D., AND SAHAI, A. Breaking the three round barrier for non-malleable commitments. In *57th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2016* (2016), IEEE.
22. GOYAL, V., LEE, C., OSTROVSKY, R., AND VISCONTI, I. Constructing non-malleable commitments: A black-box approach. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012*, pp. 51–60.

23. GOYAL, V., PANDEY, O., AND RICHELSON, S. Textbook non-malleable commitments. Cryptology ePrint Archive, Report 2015/1178, 2015. <http://eprint.iacr.org/2015/1178>, Last update: 29-Dec-2016.
24. GOYAL, V., PANDEY, O., AND RICHELSON, S. Textbook non-malleable commitments. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC* (2016), pp. 1128–1141.
25. GOYAL, V., RICHELSON, S., ROSEN, A., AND VALD, M. An algebraic approach to non-malleability. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS* (2014), pp. 41–50.
26. HAZAY, C., AND VENKITASUBRAMANIAM, M. On the power of secure two-party computation. In *CRYPTO 2016, Proceedings, Part II* (2016), vol. 9815 of *LNCS*, Springer, pp. 397–429.
27. KATZ, J., AND OSTROVSKY, R. Round-optimal secure two-party computation. In *CRYPTO 2004, Proceedings* (2004), pp. 335–354.
28. LAPIDOT, D., AND SHAMIR, A. Publicly verifiable non-interactive zero-knowledge proofs. In *Advances in Cryptology - CRYPTO* (1990).
29. LIN, H., AND PASS, R. Constant-round non-malleable commitments from any one-way function. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC* (2011), ACM, pp. 705–714.
30. LIN, H., AND PASS, R. Constant-round nonmalleable commitments from any one-way function. *J. ACM* 62, 1 (2015), 5:1–5:30.
31. LIN, H., PASS, R., AND VENKITASUBRAMANIAM, M. Concurrent non-malleable commitments from any one-way function. In *TCC* (2008), R. Canetti, Ed., vol. 4948 of *LNCS*, Springer, pp. 571–588.
32. LIN, H., PASS, R., AND VENKITASUBRAMANIAM, M. A unified framework for concurrent security: universal composability from stand-alone non-malleability. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009* (2009), pp. 179–188.
33. MAHMOODY, M., AND PASS, R. The curious case of non-interactive commitments - on the power of black-box vs. non-black-box use of primitives. In *CRYPTO 2012, Proceedings* (2012), vol. 7417 of *LNCS*, Springer, pp. 701–718.
34. A. Mittelbach and D. Venturi. Fiat-shamir for highly sound protocols is instantiable. *SCN, 2016, Proceedings*, vol. 9841 of *LNCS*, pages 198–215. Springer, 2016.
35. NAOR, M. Bit commitment using pseudorandomness. *J. Cryptology* 4, 2 (1991), 151–158.
36. OSTROVSKY, R., PERSIANO, G., AND VISCONTI, I. Constant-round concurrent non-malleable zero knowledge in the bare public-key model. In *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations* (2008), pp. 548–559.
37. OSTROVSKY, R., PERSIANO, G., AND VISCONTI, I. Simulation-based concurrent non-malleable commitments and decommitments. In *TCC 2009* (2009), vol. 5444 of *LNCS*, Springer, pp. 91–108.
38. PANDEY, O., PASS, R., AND VAIKUNTANATHAN, V. Adaptive one-way functions and applications. In *CRYPTO 2008 Proceedings* (2008), pp. 57–74.
39. PASS, R. Bounded-concurrent secure multi-party computation with a dishonest majority. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004* (2004), ACM, pp. 232–241.
40. PASS, R. Unprovable security of perfect NIZK and non-interactive non-malleable commitments. In *TCC* (2013), pp. 334–354.

41. PASS, R., AND ROSEN, A. Bounded-concurrent secure two-party computation in a constant number of rounds. In *44th Symposium on Foundations of Computer Science (FOCS 2003), Proceedings* (2003), IEEE Computer Society, pp. 404–413.
42. PASS, R., AND ROSEN, A. Concurrent non-malleable commitments. In *(FOCS (2005))*, pp. 563–572.
43. PASS, R., AND ROSEN, A. New and improved constructions of non-malleable cryptographic protocols. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005* (2005), ACM, pp. 533–542.
44. PASS, R., AND ROSEN, A. New and improved constructions of non-malleable cryptographic protocols. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005* (2005), pp. 533–542.
45. PASS, R., AND ROSEN, A. Concurrent nonmalleable commitments. *SIAM J. Comput.* *37*, 6 (2008), 1891–1925.
46. PASS, R., AND ROSEN, A. New and improved constructions of nonmalleable cryptographic protocols. *SIAM J. Comput.* *38*, 2 (2008), 702–752.
47. PASS, R., AND WEE, H. Constant-round non-malleable commitments from sub-exponential one-way functions. In *Advances in Cryptology - EUROCRYPT 2010, Proceedings* (2010), pp. 638–655.
48. ROMPEL, J. One-way functions are necessary and sufficient for secure signatures. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing* (1990), pp. 387–394.
49. SAHAI, A. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *FOCS* (1999), IEEE Computer Society, pp. 543–553.
50. SANTIS, A. D., CRESCENZO, G. D., OSTROVSKY, R., PERSIANO, G., AND SAHAI, A. Robust non-interactive zero knowledge. In *CRYPTO 2001, Proceedings* (2001), vol. 2139 of *LNCS*, Springer, pp. 566–598.
51. WEE, H. Black-box, round-efficient secure computation via non-malleability amplification. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS* (2010), IEEE Computer Society, pp. 531–540.
52. YUNG, M., AND ZHAO, Y. Generic and practical resettable zero-knowledge in the bare public-key model. In *EUROCRYPT 2007, Proceedings* (2007), vol. 4515 of *LNCS*, Springer, pp. 129–147.